



Masterarbeit

Generierung von Testdaten für Materialflusssteuerungen im Bereich der Intralogistik durch den Einsatz einer Host-Simulation

eingereicht an der

Montanuniversität Leoben

erstellt am

Lehrstuhl Industrielogistik

Vorgelegt von:

Jakob Herka, BSc
00835322

Betreuer/Gutachter:

Univ.-Prof. Mag. Dr. Helmut Zsifkovits

Leoben, am 14. November 2017

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Leoben, am _____

Datum

Unterschrift

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die mich während der Verfassung dieser Arbeit und im Laufe meines gesamten Studiums unterstützt haben.

Ein besonderer Dank gilt Herrn DI David Pietzka, der mir die Erstellung dieser Arbeit ermöglicht hat und mir immer wieder mit Rat und Tat zur Seite stand. Außerdem möchte ich mich bei Herrn DI Johannes Kapeller bedanken, dass er auf all meine Fragen eine Antwort wusste sowie bei Herrn Univ.-Prof. Mag. Dr. Helmut Zsifkovits für die Möglichkeit, diese Arbeit am Lehrstuhl Industrielogistik zu verfassen.

Vor allem gebührt mein Dank aber meinen Eltern und meiner Freundin. Meiner Mutter und meinem leider bereits verstorbenen Vater möchte ich danken, dass sie mir die Gelegenheit geboten haben, dieses Studium zu absolvieren und meiner Andrea dafür, dass sie mich unterstützt und motiviert hat, wo immer sie konnte.

Kurzfassung

Mit zunehmendem Automatisierungsgrad von Lagersystemen spielt auch das Testen im Allgemeinen und das der Performance im Besonderen eine immer größere Rolle. Die Materialflusssteuerung eines Lagersystems kommuniziert dabei im Regelfall mit übergeordneten Systemen, mit denen Daten bidirektional ausgetauscht werden. Bei der Abwicklung der Tests muss sichergestellt werden, dass auch ohne die Verfügbarkeit der übergeordneten Systeme Daten vorhanden sind. Dies kann mittels einer Simulation bewerkstelligt werden.

Die vorliegende Masterarbeit greift diese Problematik auf und verdeutlicht sie anhand eines Beispiels aus der Praxis. Zu Beginn wird der Aufbau von Materialflusssteuerungen sowie ihrer überlagerten Systeme erläutert.

Darauffolgend wird eine Definition des Begriffes „Simulation“ erstellt sowie ein Einblick in dessen Einsatzgebiete vorgenommen. Am Beispiel eines Industrieprojektes wird die Kommunikation zwischen Materialflusssteuerung und übergeordnetem System beschrieben und die darin entstandenen Anforderungen an das jeweilige Testsystem aufgelistet. Anhand dieser Vorgaben erfolgt die Erstellung einer Simulation zur Erzeugung der benötigten Testdaten, deren Funktionsumfang und Bedienung im Detail erläutert werden. Abschließend wird ein Ausblick auf mögliche weiterführende Projekte gegeben.

Abstract

The rising degree of automation in warehouse systems makes testing in general and especially testing the performance of such warehouses more and more important. Normally the material flow control of a warehouse communicates with superior systems, with which data is exchanged bidirectionally. For the execution of tests, it has to be ensured that data is available without the accessibility of these superior systems. This can be accomplished by introducing a simulation.

The present master thesis deals with this problem and illustrates it with the aid of an example of practical relevance. The structure of material flow controls and their superior systems is presented at the outset.

Furthermore, the term simulation is defined and insight into possible applications is provided. The communication between the material flow control and the superior system is shown for a specific project and all requirements for a test system are listed. The simulation for generating the test data is created on the basis of these standards. Said simulation is presented together with a detailed explanation for handling it.

In conclusion a prospect on possible additional projects is given.

Inhaltsverzeichnis

Eidesstattliche Erklärung	I
Danksagung	II
Kurzfassung	III
Abstract	IV
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	X
Abkürzungsverzeichnis	XI
1 Einleitung	1
1.1 Ausgangssituation	1
1.2 Ziel und Inhalt der Arbeit	2
2 Materialflusssteuerung und Warehouse Management	3
2.1 Definitionen	4
2.2 Allgemeiner Aufbau einer automatisierten Materialflusssteuerung	6
2.2.1 Ebenenmodell	6
2.2.2 Aufgaben der einzelnen Ebenen	7
2.2.3 Steuerungspyramide	11
2.3 Projektspezifischer Aufbau	13
3 Simulation	16
3.1 Grundlagen der Simulation	16
3.1.1 System und Modell	16

3.1.2	Weitere Definitionen	18
3.1.3	Simulationspipeline	18
3.1.4	Simulationsstudie	21
3.2	Anwendungsgebiete einer Simulation	23
3.2.1	Simulation in der Logistik	24
3.2.1.1	Planung	24
3.2.1.2	Realisierung	25
3.2.1.3	Betrieb	26
3.3	Vor- und Nachteile von Simulationen	26
4	Auswahl eines geeigneten Testsystems	29
4.1	Nachrichtenstruktur	29
4.1.1	Event	30
4.1.2	Container	31
4.1.3	FP (Function Point)	33
4.1.4	WS (Working Step)	34
4.2	Testsituationen und -mechanismen	35
4.2.1	Hinzufügen eines Eintrags - ADD	36
4.2.2	Löschen eines Eintrags - DEL	36
4.2.3	Modifizieren eines Eintrags - MOD	37
4.3	Anforderungen an ein Testsystem	37
4.3.1	Arten von Tests	38
4.3.1.1	Funktionale Tests	38
4.3.1.2	Massentests	39
4.3.2	Arten von Nachrichten	39
4.3.2.1	Ankündigung eines Wareneingangs	39
4.3.2.2	Widerrufung eines Wareneingangs	40
4.3.2.3	Ankündigung eines leeren Behälters	40
4.3.2.4	Ankündigung eines neuen Kommissionierauftrags	41
4.3.2.5	Stornierung eines Kommissionierauftrags	41
4.3.2.6	Modifizierung eines Kommissionierauftrags	41
4.3.2.7	Anforderung einer Rücklieferung	42

4.3.2.8	Anforderung der Rücklieferung bei Saisonende	42
4.3.2.9	Widerrufung einer Rücklieferung	42
4.3.2.10	Hinzufügen einer Versandrampe	43
4.3.2.11	Bestätigung einer Versandrampe	43
4.3.2.12	Ankündigung neuer Artikeldaten	43
4.3.2.13	Sperren/Entsperren eines Artikels	44
4.4	Mögliche Lösungsansätze	44
4.4.1	Manuelle Datenerstellung	44
4.4.1.1	Erstellung mittels Textfile	44
4.4.1.2	Erstellung mittels MS Excel	45
4.4.2	Automatische Datengenerierung - Einsatz einer Simulation	45
5	Durchführung der Tests mittels Simulation	47
5.1	Implementierung der Host-Simulation	47
5.1.1	Erstellung der nötigen Tabellen	48
5.1.2	Implementierung der Prozeduren	49
5.1.3	Erstellung der Benutzeroberfläche	51
5.2	Bedienung der Simulation	52
5.2.1	Erstellung eines Szenarios	52
5.2.2	Darstellung der Simulationsdaten	57
5.2.3	Anzeige der Artikeltabellen	63
5.3	Simulationsdurchlauf anhand eines konkreten Beispiels	65
6	Conclusio und Ausblick	72
	Literaturverzeichnis	XII
	Anhang	XVII

Abbildungsverzeichnis

2.1	Ebenenmodell für Materialflusssteuerungen	8
2.2	Vereinfachte Steuerungspyramide zentraler Materialflusssteuerungen . .	12
2.3	Steuerungspyramide des projektspezifischen Aufbaus	14
3.1	Methoden zur Untersuchung eines Systems	17
3.2	Simulationspipeline	19
3.3	Durchführung einer Simulationsstudie	22
4.1	Aufbau einer Nachricht vom Host ans WCS	30
5.1	Ansicht der Szenariokonfiguration	54
5.2	Kontextmenü in der Szenariokonfiguration	55
5.3	Dialog zur Erstellung eines neuen Szenarios	55
5.4	Dialog, um einen Parameter zu einem Szenario hinzuzufügen	56
5.5	Kontextmenü für Parameter in der Szenariokonfiguration	57
5.6	Ansicht der Simulationsdaten	58
5.7	Filter zur Einschränkung der Anzeige der Simulationsdaten	59
5.8	Kontextmenü für Events	59
5.9	Kontextmenü für Container	60
5.10	Eingabedialog zur Aktualisierung eines Kommissionierauftrags	61
5.11	Eingabedialog zum Hinzufügen einer Rampe	61
5.12	Ansicht der Simulationsdaten für Nachrichten ohne Container-Eintrag .	62
5.13	Kontextmenü für Artikel	63
5.14	Artikeltabelle	64
5.15	Erstellung eines Szenarios	65
5.16	Erstellung eines Parameters für die Ankündigung eines Kommissionier- auftrags	66

5.17 Szenario-Parameter für die Ankündigung eines Kommissionierauftrags . . .	67
5.18 Container-Parameter für die Ankündigung eines Kommissionierauftrags . . .	68
5.19 SKU-Parameter für die Ankündigung eines Kommissionierauftrags . . .	68
5.20 Simulationsdaten für die Ankündigung eines Kommissionierauftrags . . .	70

Tabellenverzeichnis

3.1	Vor- und Nachteile von Simulationen	28
4.1	Felder des Event-Eintrags	31
4.2	Felder des Container-Eintrags	32
4.3	Felder des FP-Eintrags	33
4.4	Felder des WS-Eintrags	34

Abkürzungsverzeichnis

GUI	Benutzerinterface bzw. Graphical User Interface
ERP	Enterprise Resource Planning
WMS	Warehouse Management System
WWS	Warenwirtschaftssystem
MIS	Managementinformationssystem
PPS	Produktionsplanung und -steuerung
MFR	Materialflussrechner
WCS	Warehouse Control System
LVR	Lagerverwaltungsrechner
UST	Unterlagerte Steuerungen
SPS	Speicherprogrammierbare Steuerungen
OSR	Order Storage & Retrieval
FP	Function Point
WS	Working Step
SKU	Stock Keeping Unit
VDMA	Verband Deutscher Maschinen- und Anlagenbau
VDI	Verein Deutscher Ingenieure

1 Einleitung

1.1 Ausgangssituation

Die logistischen Anforderungen an Unternehmen sind in den letzten Jahrzehnten kontinuierlich gestiegen. Kürzere Bestellzyklen, kleinere Bestellmengen sowie eine größere Produktvarianz führen zur Notwendigkeit individueller Lager- und Kommissioniersysteme. Weitere Einflussfaktoren, wie beispielsweise steigende Personalkosten begünstigen, dass immer mehr Lager einen hohen Automatisierungsgrad aufweisen. Dies führt dazu, dass auch das Testen der Performance solcher Lager eine immer größere Bedeutung gewinnt.

Da die Materialflusssteuerungen von Lagersystemen ihre Auftragsdaten meist von übergeordneten Systemen erhalten, müssen für den Testvorgang die benötigten Daten erzeugt werden, was oft mit einem großen Aufwand verbunden ist. Diese Problematik wird in der vorliegenden Arbeit aufgegriffen und anhand eines praxisrelevanten Beispiels verdeutlicht. Konkret geht es um die Erstellung von Testdaten für ein neues Lager eines Sportartikelherstellers. Das neue Lager soll ausschließlich für den Onlinehandel genutzt werden und in Kombination mit einem bestehenden Distributionslager, das für die Verteilung der Waren an die einzelnen Filiallager zuständig ist, den Absatzmarkt bedienen. Dieses wird die Waren auch an das neu errichtete E-Commerce-Lager liefern.

Im Zuge der Inbetriebnahme dieses Lagers sollen mehrere Testreihen abgeführt werden, welche die Bereitschaft von Software und Materialfluss beweisen sollen. Für diesen Zweck müssen große Mengen an auftragsbezogenen Testdaten erstellt werden. Dieser Prozess kann im Normalfall auf verschiedene Arten durchgeführt werden. Manuelle und halbautomatische Verfahrensschritte führen im konkreten Fall aber zu einem sehr großen und teilweise schwer zu bewältigenden Zeitaufwand. Außerdem ist für einige der Prozessschritte eine bidirektionale Kommunikation der Lagersoftware zu überge-

ordneten Host-Systemen notwendig. Diese muss für die gewünschten Tests durch eine Simulation entsprechend nachgebaut werden.

1.2 Ziel und Inhalt der Arbeit

Ziel dieser Arbeit ist die Erstellung eines Simulationsmodells für die Testumgebung, um ein schnelles und effizientes, aber auch benutzerfreundliches Testen zu ermöglichen. Ein Benutzerinterface bzw. Graphical User Interface (GUI) soll dem Anwender eine einfache Bedienung der Simulation ermöglichen. In dieser Arbeit werden zuerst die notwendigen theoretischen Hintergründe erläutert und anschließend die Problematik an einem praxisrelevanten Beispiel verdeutlicht.

Kapitel 2 soll einen Überblick über den allgemeinen Aufbau von Lagersystemen und Materialflusssteuerungen geben. Dies schafft die Voraussetzung zum Verständnis der Kommunikation zwischen einzelnen Komponenten dieser Systeme. Am Ende dieses Kapitels wird der Aufbau des konkreten Projekts erläutert.

Anschließend folgt eine Erläuterung des Begriffs Simulation sowie damit verbundener Begriffe in Kapitel 3. Nach der Darlegung von Grundlagen zu dieser Thematik werden notwendige Schritte zur erfolgreichen Anwendung einer Simulation vorgestellt. Auch auf verschiedene Einsatzgebiete von Simulationen, vor allem im Bereich der Logistik, wird Bezug genommen.

Das 4. Kapitel behandelt die Kommunikation zwischen Lagersoftware und übergeordneten Systemen sowie die daraus resultierenden Anforderungen an ein geeignetes Testsystem. Unterschiedliche Lösungsansätze werden miteinander verglichen und Vor- und Nachteile der einzelnen Ansätze beschrieben.

Kapitel 5 erläutert die im Rahmen der Arbeit erstellte und implementierte Simulation. Des Weiteren wird die Benutzeroberfläche gezeigt sowie die Bedienung der Software erklärt. Auf Basis eines Testszenarios werden die Vorteile, welche durch die erstellte Simulation errungen werden konnten, dargelegt.

Zum Abschluss zeigt der Autor in Kapitel 6 Handlungsempfehlungen für zukünftige Projekte und Forschungsfragen.

2 Materialflusssteuerung und Warehouse Management

In diesem Kapitel wird der systematische Aufbau einer Materialflusssteuerung in einem Lager bzw. Lagersystem erläutert. Dabei werden die Begriffe „Materialflusssteuerung“ und „Warehouse Management“ einer näheren Betrachtung unterzogen. Weiters wird der projektspezifische Aufbau im konkreten Fall betrachtet und mit dem theoretischen Modell verglichen.

Die Bedeutung des Begriffs der Materialflusssteuerung hat sich im Laufe der Zeit verändert. Früher bezeichnete sie hauptsächlich Einrichtungen zum Transport von Ladungseinheiten von einem Ort zum anderen. Heute stellen Materialflusssteuerungen jedoch komplexe und optimierte Systeme dar, in deren Aufgabenbereich die Steuerung, Visualisierung und Kontrolle des Materialflusses fallen. Sie übernehmen neben grundlegenden Aufgaben, wie dem Ausführen der Transportaufträge, auch weiterführende Funktionen, wie Ressourcenoptimierung. Ein weiterer wichtiger Punkt ist die Erfassung von Daten und deren Weitergabe an andere Systeme, meist an ein übergeordnetes Enterprise Resource Planning (ERP) System.¹

Der Aufbau heutiger Materialflusssteuerungen ist hierarchisch, pyramidenförmig und kann anhand mehrerer Ebenen beschrieben werden. Auf das Ebenenmodell für Materialflusssteuerungen, welches auch die übergeordneten Systeme miteinschließt, wird in Kapitel 2.2.1 näher eingegangen.²

Warehouse Management ist ein Begriff aus dem Englischen, für welchen kein gleich-

¹Vgl. Nieke 2010, S.15 ff.

²Vgl. Nieke 2010, S.17.

bedeutendes Pendant in der deutschen Sprache gefunden werden kann. Die wörtliche Übersetzung „Lagerverwaltung“ wird der gesamten Bedeutung des englischen Ausdrucks nicht gerecht. In Ten Hompel und Schmidt³ wird Warehouse Management folgendermaßen definiert: „Das Warehouse Management bezeichnet im allgemeinen Sprachgebrauch [...] die Steuerung, Kontrolle und Optimierung komplexer Lager- und Distributionssysteme. Neben den elementaren Funktionen einer Lagerverwaltung wie Mengen- und Lagerplatzverwaltung, Fördermittelsteuerung und -disposition gehören nach dieser Betrachtungsweise auch umfangreiche Methoden und Mittel zur Kontrolle der Systemzustände und eine Auswahl an Betriebs- und Optimierungsstrategien zum Leistungsumfang.“ Im Folgenden wird der Begriff Warehouse Management als übergreifendes Instrumentarium verwendet, welches sowohl die Materialflusssteuerung als auch deren übergeordnete Systeme miteinschließt.⁴

2.1 Definitionen

Im Gegensatz zum bereits beschriebenen Begriff Warehouse Management, der die Materialflusssteuerung miteinschließt, bezeichnet ein Warehouse Management System (WMS) ein der Materialflusssteuerung überlagertes System. Dieses beschäftigt sich mit der Führung und Optimierung innerbetrieblicher Lagersysteme und hat somit viele Schnittstellen zu angrenzenden Systemen. Im Folgenden sollen kurz ein paar solcher Systeme vorgestellt werden, wobei natürlich je nach Anwendungsfall nicht immer alle der genannten Systeme eingesetzt werden. Außerdem können einige davon situationsabhängig auch Bestandteil eines WMS sein.⁵

Zum besseren Verständnis sind vorab folgende Definitionen zu treffen:

- „*Warenwirtschaftssystem (WWS)*, EDV-System zur Unterstützung der gesamten Auftragsabwicklung und Warenwirtschaft im Unternehmen. Im WWS werden z.B. Kundenaufträge erfasst, die Warenbewegungsdaten gebucht, Bestände verwaltet, Rechnungen erzeugt usw. Bei Betrieben (z.B. Handel) mit einem großen Lager hat das WWS eine Verbindung zu einem separaten Lagerverwaltungssystem. In

³Ten Hompel und Schmidt 2010, S.8.

⁴Vgl. Ten Hompel und Schmidt 2010, S.8.

⁵Vgl. Ten Hompel und Schmidt 2010, S.9.

diesem Fall erfolgt die Bestandverwaltung und die Steuerung der Lagerprozesse (z.B. Kommissionieraufträge) im Lagerverwaltungssystem. Über eine Schnittstelle werden Daten in das WWS zurückgemeldet bzw. übergeben.⁶

- „*Managementinformationssystem* (abgek. MIS) hat als vorrangige Aufgabe die Vorbereitung von Managemententscheidungen. MIS werden oftmals als Bestandteil eines WWS geführt. Seit Mitte der 90er Jahre werden zunehmend analytische Funktionen in MIS integriert. Trends, Prognosen und Analysen im echtzeitnahen Bereich sollen das Management unterstützen.“⁷
- „*Produktionsplanung und -steuerungssystem* (abgek. PPS, engl. Production planning and control system) umfasst informationsverarbeitende Systeme der Produktionsplanung und -steuerung. PPS-Systeme lassen sich nach dem Steuerungsprinzip [...] einteilen.“⁸
- „*ERP-System*, Abkürzung für Enterprise Resource Planning System; Sammelbegriff für Software-Systeme mit breiter Funktionalität, die den gesamten Prozess der betrieblichen Leistungserstellung materiell und monetär abbilden.“⁹
Ein ERP-System „... ist ein integriertes Softwaresystem zur umfassenden Planung und Koordination unternehmerischer, insbesondere betriebswirtschaftlicher Aufgaben mit dem Ziel, die in einem Unternehmen vorhandenen Ressourcen möglichst effizient einzusetzen.“¹⁰
- „*Materialflussrechner (MFR)*, wesentlicher Bestandteil eines Lagerverwaltungssystems, der die Umwandlung und Abwicklung von Transportaufträgen übernimmt. Der M. organisiert und strukturiert den Transport aller Waren im Lager und gibt Signale an die Steuerungen der Lager- und Fördertechnik.“¹¹
- „*Warehouse Control System* (abgek. WCS) Vergleichbar mit dem MFR kontrollieren WCS Quelle-Ziel-Beziehungen. Typischerweise werden zusätzliche Aufgaben

⁶Bichler et al. 2005, S.195.

⁷Ten Hompel und Heidenblut 2006, S.142.

⁸Ten Hompel und Heidenblut 2006, S.177.

⁹Klaus und Krieger 2008, S.169 ff.

¹⁰Ten Hompel und Heidenblut 2006, S.63 ff.

¹¹Bichler et al. 2005, S.145.

integriert, die über den Umfang eines reinen MFR hinausgehen. WCS können insbesondere lokale bzw. nicht bewegte Bestände verwalten. Sie gelangen insbesondere dort zum Einsatz, wo wesentliche Funktionen eines WMS durch WWS bzw. ERP-Systeme abgedeckt werden und demzufolge ein separates WMS nicht erforderlich ist.“¹²

2.2 Allgemeiner Aufbau einer automatisierten Materialflusssteuerung

Wie bereits am Beginn dieses Kapitels erwähnt wurde, können heutige Materialflusssteuerungen in mehrere Ebenen bzw. Schichten unterteilt werden. Der folgende Abschnitt beschreibt den Aufbau eines solchen Modells.

2.2.1 Ebenenmodell

Mit Hilfe eines Ebenenkonzepts soll eine Aufgabenverteilung erreicht werden, wobei auf einer Schicht jeweils ähnliche Aufgaben bearbeitet werden. Auch der Datenaustausch erfolgt im Wesentlichen innerhalb einer Ebene und Schnittstellen zwischen den Ebenen müssen klar definiert sein.¹³

Krämer¹⁴ definiert folgende Strukturierungsprinzipien für die Ebenenbildung:

- In einer Ebene werden ähnliche Aufgaben bearbeitet.
- Eine Ebene bearbeitet allgemein verwendbare, abstrakt beschreibbare Aufgaben.
- Die Aufgaben sind möglichst vollständig und autark zu bearbeiten. Es dürfen keine impliziten Informationen anderer Ebenen genutzt werden.
- Kommunikation erfolgt hauptsächlich auf einer Ebene.
- Die Bindung zu über- oder unterlagerten Ebenen sei hauptsächlich auf den Austausch von Befehlen und Ergebnissen beschränkt.

¹²Ten Hompel und Schmidt 2010, S.9 ff.

¹³Vgl. Krämer 2002, S.29 ff.

¹⁴Krämer 2002, S.30.

- Befehle über mehrere Ebenen hinweg sind verboten.
- Die nächsthöhere Ebene abstrahiert die erhaltenen Informationen und gibt abstrahierte Befehle an die unterlagerte Ebene.
- Daten unterer Ebenen können transparent von höheren Ebenen gesehen werden.

Auf Grund dieser Strukturierungsprinzipien kann nun ein Ebenenmodell aufgebaut werden. Speziell auf die Materialflussautomatisierung zugeschnitten wurde vom Verband Deutscher Maschinen- und Anlagenbau (VDMA) ein Ebenenmodell für Materialflusssteuerungen aufgestellt, welches in Abbildung 2.1 dargestellt ist.

2.2.2 Aufgaben der einzelnen Ebenen

Die einzelnen Schichten dieses Ebenenmodells sollen in diesem Kapitel genauer erläutert werden:

- **Warenwirtschafts-/Produktionsplanungssystem (WWS/PPS):**
Diese Ebene dient der Darstellung des IT-Modells aller wirtschaftlichen Gegebenheiten und Prozesse eines Betriebs. Hier erfolgt die Erfassung der Kundenaufträge und die anschließende Übermittlung der notwendigen Daten an die darunterliegende Lagerverwaltung. Genauere Definitionen für WWS und PPS erfolgten bereits in Kapitel 2.1.¹⁵
- **Lagerverwaltung:**
Die Lagerverwaltung wird durch ein WMS durchgeführt und stellt dessen Hauptaufgabe dar. Sie verwaltet Orte, Bereiche, Topologie und Belegung und organisiert die ablaufenden Prozesse. Das bedeutet, es werden sowohl die vorhandenen Lagerplätze in einem System (Platzverwaltung) als auch die darin gelagerten Einheiten (Bestandsverwaltung) kontrolliert und dokumentiert. Im Gegensatz zu ihrer übergeordneten Ebene beschäftigt sich die Lagerverwaltung nicht mit den wirtschaftlichen, sondern mit den technischen Gegebenheiten und Prozessen in einem Lager. Zwischen WMS und WWS kann allerdings schwer eine klare Abgrenzung

¹⁵Vgl. Ten Hompel und Schmidt 2006, S.S.225; Bichler et al. 2005, S.195.

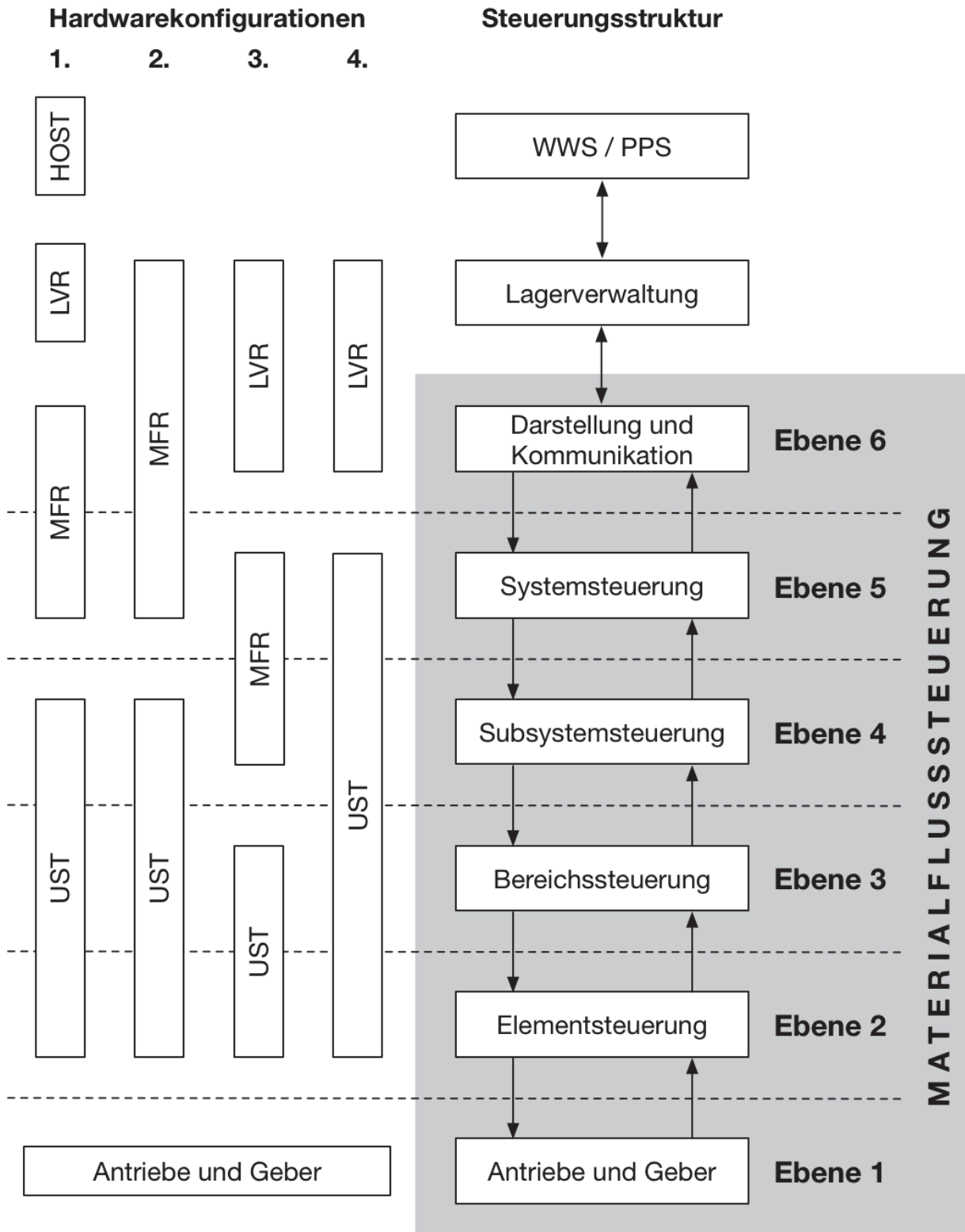


Abbildung 2.1: Ebenenmodell für Materialflusssteuerungen¹⁶

¹⁶Vgl. Jünemann und Beyer 1998, S.145 nach VDMA Einheitsblatt 15276

gezogen werden. Im Normalfall kümmert sich das WWS um Kundenaufträge und leitet diese weiter ans WMS, welches somit keine direkte Sicht auf den Markt benötigt. Durch das WMS werden diese Aufträge aufbereitet und einzelnen Zonen des Lagers zugeordnet. Von Fall zu Fall können die Aufgabenbereiche der beiden Systeme jedoch variieren.¹⁷

▪ **Darstellung und Kommunikation:**

Abbildung 2.1 zeigt, dass diese Schicht die höchste der Materialflusssteuerung ist und somit die Schnittstelle zum übergeordneten System bildet. Die eingesetzten Rechnersysteme in dieser Ebene sind Materialflussrechner, welche als einzelner Rechner oder als Verbund mehrerer Rechner in Client-Server-Architektur als Materialflussleitsystem fungieren. Dieses übernimmt die übermittelten Transportanforderungen vom übergeordneten System und wandelt sie in Transportaufträge mit Systemkoordination für die Systemsteuerung um. An diese werden jene anschließend weitergeleitet. Außerdem erfolgt über das Materialflussleitsystem die Rückmeldung an die Lagerverwaltung in Form von Auftragsvollzugs- oder Störmeldungen. Weitere Funktionen dieser Ebene, wie die Anlagenbedienung, die Visualisierung des Anlagenzustands sowie die Protokollierung der Abläufe, ermöglichen eine Entkopplung des Materialflusssystems von der überlagerten Ebene und lassen einen teilautomatischen Betrieb zu.¹⁸

▪ **Systemsteuerung:**

Wie bereits der Name dieser Ebene deutlich macht, ist sie zuständig für die Steuerung sämtlicher Transportoperationen des Materialflusssystems. Die Realisierung erfolgt dabei meist auch auf dem Materialflussleitsystem. Erhaltene Transportanforderungen werden hier als Aufträge eingelastet, um anschließend verteilt zu werden. Um die hierfür benötigten Informationen bereitzustellen, enthält diese Ebene ein Modell des Materialflusssystems mit allen Topologiedaten, Auftragsbeständen und Statusinformationen. Außerdem sind in der Systemsteuerung Transportstrategien implementiert, welche den optimalen Weg für den Transport je nach Anlagenzustand vorgeben. Hier befindet sich die Zentralsteuerung des Ma-

¹⁷Vgl. Ten Hompel und Schmidt 2010, S.10; S.54; S.225.

¹⁸Vgl. Ten Hompel und Schmidt 2006, S.163 ff.; Jünemann und Beyer 1998, S.144 ff.

terialflusssystemen, die mehrere Subsystemsteuerungen unterlagert sind. An diese werden die Transportaufträge, nach der Verteilung auf die dafür einzusetzenden Fördersysteme, weitergeleitet.¹⁹

▪ **Subsystemsteuerung:**

Wie bereits erwähnt und in Abbildung 2.1 dargestellt gibt es auf dieser Ebene mehrere Subsystemsteuerungen, auch unterlagerte Steuerungen (UST) genannt, jeweils eine für ein abgeschlossenes Teilsystem der Gesamtanlage. Alle Operationen dieser Teilsysteme werden dezentral von einer UST gesteuert. Beispiele für solche Teilsysteme sind etwa ein Hochregallager oder ein Fahrerloses Transportsystem, aber auch ein Verbund solcher Fahrzeuge. Realisiert werden die Subsystemsteuerungen durch Industrie-PCs oder Speicherprogrammierbare Steuerungen (SPS) der oberen Leistungsklasse. Um Inbetriebnahme und Wartungsarbeiten zu erleichtern wird durch Bedienungs- und Visualisierungsfunktionen der teilautomatische Betrieb einzelner Teilsysteme ermöglicht.²⁰

▪ **Bereichssteuerung:**

Die Subsystemsteuerung kann wiederum in einzelne funktionale Bereiche unterteilt werden. In diesen finden sich eigene Bereichssteuerungen, die den Betrieb einzelner Fördermittel kontrollieren. Beispielsweise gibt es in einem Verbund von Fahrerlosen Transportfahrzeugen eine Bereichssteuerung für ein einzelnes Fahrzeug oder ein Abschnitt einer Förderstrecke wird separat gesteuert. Realisiert wird sie hardwaretechnisch entweder als SPS der mittleren Leistungsklasse oder als Industrie-PC. Die Aufgabe der Bereichssteuerung ist die Platzverwaltung auf den Fördermitteln, da ein solches meist mehrere Förderplätze hat, die von transportierten Ladeeinheiten belegt werden können. Durch verschiedene Verfahren wird hier der Materialfluss gesteuert und überwacht, z. B. werden auf einer Förderstrecke Behälter vermerkt, die zuvor von einem Barcodescanner identifiziert wurden.²¹

¹⁹Vgl. Ten Hompel und Schmidt 2006, S.164; Jünemann und Beyer 1998, S.145 ff.

²⁰Vgl. Ten Hompel und Schmidt 2006, S.164; Jünemann und Beyer 1998, S.146.

²¹Vgl. Ten Hompel und Schmidt 2006, S.164; Jünemann und Beyer 1998, S.146.

▪ **Elementsteuerung:**

Eine Bereichssteuerung führt einzelne Elementsteuerungen, welche wiederum ihrerseits für die Führung eines oder mehrerer Antriebe eines Fördermittelplatzes zuständig sind. Diese werden als Bestandteil der Bereichssteuerung, mit einer selbstständigen Klein-SPS oder als Mikrocontroller ausgeführt. Für die Antriebe werden hier typische Abläufe und Sollwerte generiert und überwacht. Außerdem werden auf dieser Ebene Sicherheitsmechanismen durchgeführt und es erfolgt eine Synchronisation mit benachbarten Elementsteuerungen.²²

▪ **Antriebe und Geber:**

Antriebe (Aktoren) und Geber (Sensoren) stellen die unterste Ebene einer Materialflusssteuerung dar und bilden die Schnittstelle zur darunter liegenden Materialflussebene. Erstere liefern die mechanische Energie für Transportbewegungen. Die Führung der Antriebssteuerung durch die überlagerte Ebene erfolgt je nach Antrieb durch analoge oder digitale Signale. Messwerte über den Zustand des Systems und der darin ablaufenden Prozesse werden von den Sensoren an die Elementsteuerung übermittelt.²³

2.2.3 Steuerungspyramide

In Nieke²⁴ findet sich eine Aufarbeitung des Ebenenmodells für Materialflusssteuerungen in Form einer Pyramide, welche nur die steuerungstechnisch relevanten Schichten beschreibt. Abbildung 2.2 zeigt die Pyramide mit ihren verschiedenen Ebenen und den jeweils dazugehörigen Hardwarekomponenten. Die Materialflusssteuerung umfasst die drei unteren Schichten, welche hier gelb dargestellt sind. Darunter findet sich noch die Mechanik der Transportgrundlage, auf welcher die Pyramide aufbaut und die üblicherweise nicht dargestellt wird. Im Anschluss folgt eine kurze Beschreibung der einzelnen Ebenen:

²²Vgl. Jünemann und Beyer 1998, S.146.

²³Vgl. Ten Hompel und Schmidt 2006, S.165; Jünemann und Beyer 1998, S.146.

²⁴Vgl. Nieke 2010, S.17.

▪ **Feldebene:**

Die unterste Ebene der Pyramide entspricht in etwa den Schichten „Elementsteuerung“ und „Antriebe und Geber“ des Ebenenmodells aus Kapitel 2.2.2. Dies ist der ausführende Teil der Materialflusssteuerung, bestehend aus der Mechanik, Aktoren und Sensoren.²⁵

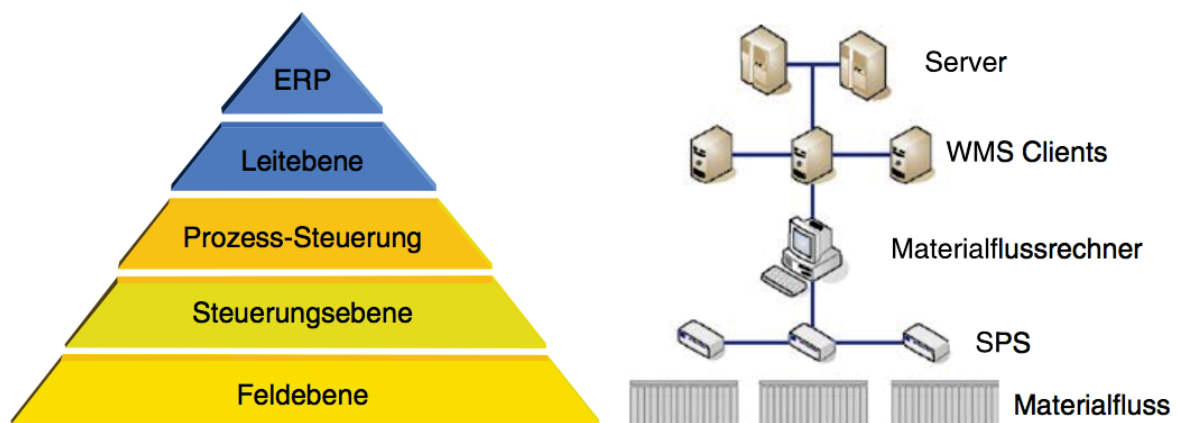


Abbildung 2.2: Vereinfachte Steuerungspyramide zentraler Materialflusssteuerungen nach Nieke²⁶

▪ **Steuerungsebene:**

Die nächsthöhere Schicht ist für die Koordination zwischen den einzelnen Transportelementen zuständig. Sie steuert Transporte und Lastübergaben und verarbeitet außerdem die Signale der Feldebene. Ihre Realisierung erfolgt meist durch den Einsatz von SPS, in welchen die nötigen Steuerungsfunktionen als Programmcode hinterlegt sind. Aus prozesstechnischer Sicht ist diese Ebene dafür zuständig, das mechanische Layout in ein logisches und platzbezogenes Transportsystem zu überführen, um automatisierte Teiltransporte zu ermöglichen. Verglichen mit dem Ebenenmodell aus Kapitel 2.2.2 umfasst diese Schicht die Ebenen der „System-, Subsystem- und Bereichssteuerung“.²⁷

²⁵Vgl. Nieke 2010, S.17.

²⁶Quelle: Nieke 2010, S.17.

²⁷Vgl. Ten Hompel und Schmidt 2006, S.165; Nieke 2010, S.17 ff.

▪ **Prozess-Steuerung:**

Dies ist das Pendant zur Ebene „Darstellung und Kommunikation“ aus Kapitel 2.2.2. Die Prozess-Steuerung wird häufig auch, wie die eingesetzten Systeme, Materialflussrechner genannt. Transportaufträge werden von hier an die Steuerungsebene gesendet, welche je nach Erfolg der Warenverschiebung eine entsprechende Rückmeldung liefert. Auch die Visualisierung ist in dieser Schicht angeordnet und es erfolgt die Kommunikation nach oben mit der Leitebene.²⁸

Die beiden obersten Ebenen können gleichgesetzt werden mit den höchsten Schichten aus Kapitel 2.2.2. Die Leitebene übernimmt eine funktionale Steuerungsaufgabe. Sie kümmert sich um die lagerverwaltungstechnischen Aufgaben und steuert auch, falls vorhanden, die Kommissionierung. Die Steuerungsaufgabe des ERP-Systems ist strategischer Natur. In seinen Verantwortungsbereich fällt vor allem die Abbildung höherwertiger Geschäftsprozesse.²⁹

2.3 Projektspezifischer Aufbau

In diesem Abschnitt folgt eine kurze Beschreibung des Aufbaus am konkreten Beispiel, für das eine Host-Simulation geschaffen werden soll. Abbildung 2.3 zeigt die Steuerungspyramide mit den einzelnen Schichten des projektspezifischen Aufbaus. Die unteren Ebenen der Materialflusssteuerung sind so aufgebaut, wie es in Kapitel 2.2 beschrieben wurde. Die Aufgaben der obersten Schicht, also der Prozesssteuerung, werden durch ein WCS ausgeführt. Die Leitebene darüber ist in 2 separate Schichten unterteilt. Sie besteht aus einem System, das in Abbildung 2.3 als „WCS_alt“ bezeichnet wird, und einem diesem überlagerten WMS. Der Grund dieser Unterteilung sowie der Bezeichnung „WCS_alt“ ist, dass die beiden Systeme bei bereits bestehenden Lagern im Einsatz sind, für welche das „WCS_alt“ die Funktionen eines WCS ausführt. Für das neue Lager wurde eine neues WCS eingeführt, wofür eine Schnittstelle zum bestehenden System geschaffen werden musste. Da der Aufbau einer Schnittstelle zwischen WCS und „WCS_alt“ mit weit weniger Aufwand verbunden war als zwischen WCS und

²⁸Vgl. Nieke 2010, S.18.

²⁹Vgl. Verriet und Van Wijngaarden 2011, S.17 ff.; Nieke 2010, S.S.18.

Generierung von Testdaten für Materialflusssteuerungen im Bereich der Intralogistik durch den Einsatz einer Host-Simulation

Kapitel 2. Materialflusssteuerung und Warehouse Management

WMS, übernimmt das „WCS_alt“ hier nur eine Durchreichfunktion und übermittelt die Daten, die es vom WMS erhält, ans WCS der Prozesssteuerung.³⁰

Auf oberster Ebene befindet sich wie üblich ein ERP-System. Die der Materialflusssteuerung überlagerten Schichten, also das „WCS_alt“, das WMS und das ERP-System, werden in dieser Arbeit fortan als Host bezeichnet. Die geforderte Host-Simulation soll genau diese Ebenen simulieren und eine Kommunikation mit dem WCS ermöglichen, was aber in Kapitel 3 genauer erklärt wird. Für das WCS soll dabei kein Unterschied im Vergleich zur Kommunikation mit den tatsächlichen Systemen entstehen.³¹

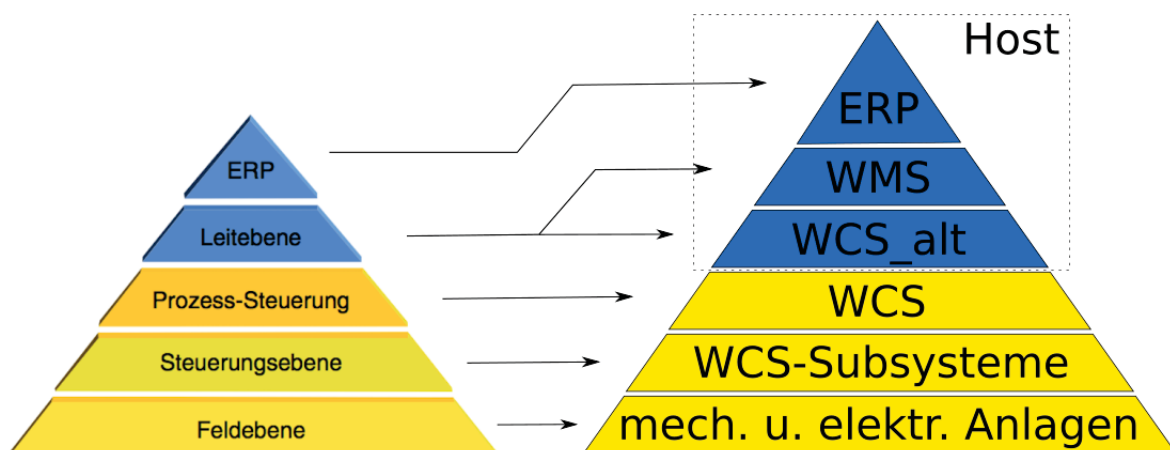


Abbildung 2.3: Steuerungspyramide nach Nieke³² mit Bezug auf den projektspezifischen Aufbau³³

Vom Host-System werden Kommissionier- und Replenishmentaufträge ans WCS weitergeleitet. Außerdem verarbeitet und bewertet es die Informationen, die es vom WCS nach Durchführung dieser Aufträge erhält.³⁴

³⁰Interne Dokumente

³¹Interne Dokumente

³²Vgl. Nieke 2010, S.17.

³³Eigene Abbildung

³⁴Interne Dokumente

Das WCS ist verantwortlich für die Kommunikation mit dem Host und mit den WCS-Subsystemen, welche die tatsächliche Bearbeitung der Aufträge vornehmen. Die Auftragsinformationen sowie die vorgesehene Route müssen in eine Form konvertiert werden, die von den Subsystemen verarbeitet werden kann. Diese steuern die mechanischen und elektrischen Anlagen, wie Motoren oder Förderbänder, und verbinden diese mit dem WCS.³⁵

Abschließend sollen noch ein paar Hinweise zum Aufbau des Lagers gegeben werden. Bereits in Kapitel 1.1 wurde darauf hingewiesen, dass das neu erstellte Lager von einem bereits vorhandenen Distributionslager mit Artikeln beliefert wird. Die Lagerung dieser Artikel erfolgt in einem OSR-System (Order Storage & Retrieval System). Dies ist ein halbautomatisiertes Kommissionier- und Lagersystem, welches mittels Lift und Shuttle die Artikel aus einem Hochregallager in Behälter ein- und auslagert. Über ein System von Förderern werden die Behälter anschließend zu Versandrampen transportiert, wo sie entsprechend vom Host übermittelter Vorgaben ausgeschleust werden.³⁶

³⁵Interne Dokumente

³⁶Interne Dokumente

3 Simulation

Dieses Kapitel legt ganz allgemein dar, was Simulation ist, wie sie funktioniert und wo und wie sie eingesetzt werden kann. Dies soll zeigen, warum Simulation ein guter Ansatz zur Lösung des vorliegenden Problems ist. Zu Beginn wird der Begriff Simulation selbst genauer beleuchtet.

3.1 Grundlagen der Simulation

Der Verein Deutscher Ingenieure (VDI) hat eine Richtlinie zum Thema Simulation von Logistik-, Materialfluss- und Produktionssystemen herausgegeben. Hier wird Simulation als ein Verfahren zur Nachbildung eines Systems definiert. Unter Berücksichtigung seiner dynamischen Prozesse wird das System in einem experimentierbaren Modell abgebildet. Das Ziel ist es, daraus Erkenntnisse zu gewinnen, um sie anschließend auf die Wirklichkeit zu übertragen. Fasst man den Begriff Simulation weiter, so beinhaltet er die Vorbereitung, Durchführung und Auswertung geplanter Experimente, was mit einem Simulationsmodell bewerkstelligt wird. Dies ermöglicht die Untersuchung des zeitlichen Ablaufverhaltens komplexer Systeme.³⁷

3.1.1 System und Modell

In obiger Definition finden sich die Begriffe System und Modell. Hall und Fagen³⁸ definieren den Begriff System folgendermaßen: „Ein System ist eine Menge von Objekten, gemeinsam mit den Beziehungen zwischen den Objekten und zwischen ihren Eigenschaften.“ Demnach reicht es also nicht aus ein System als eine Ansammlung einzelner Bestandteile zu sehen, es geht vielmehr auch um die Wechselwirkungen der einzelnen

³⁷Vgl. Daniluk und Chisu 2010, S.147.

³⁸Hall und Fagen 1956, S.18.

Elemente und wie sie sich gegenseitig beeinflussen. Um ein solches System abzubilden bedarf es eines Modells, welches das System auf seine wesentlichen Bestandteile und Beziehungen reduziert und damit die Komplexität verringert.³⁹

Ein solches Modell ist von Vorteil, wenn ein komplexes System untersucht werden soll, da Experimente am Modell anstatt am realen System durchgeführt werden können. Abbildung 3.1 zeigt unterschiedliche Methoden zur Untersuchung eines Systems. Die Ausführung des Modells kann in physischer Form erfolgen, was aber einen hohen Kostenaufwand mit sich bringt. Alternativ besteht die Möglichkeit, ein mathematisches Modell zu erstellen. Dieses kann anschließend mittels analytischer Methoden oder aber mittels einer Simulation untersucht werden.⁴⁰

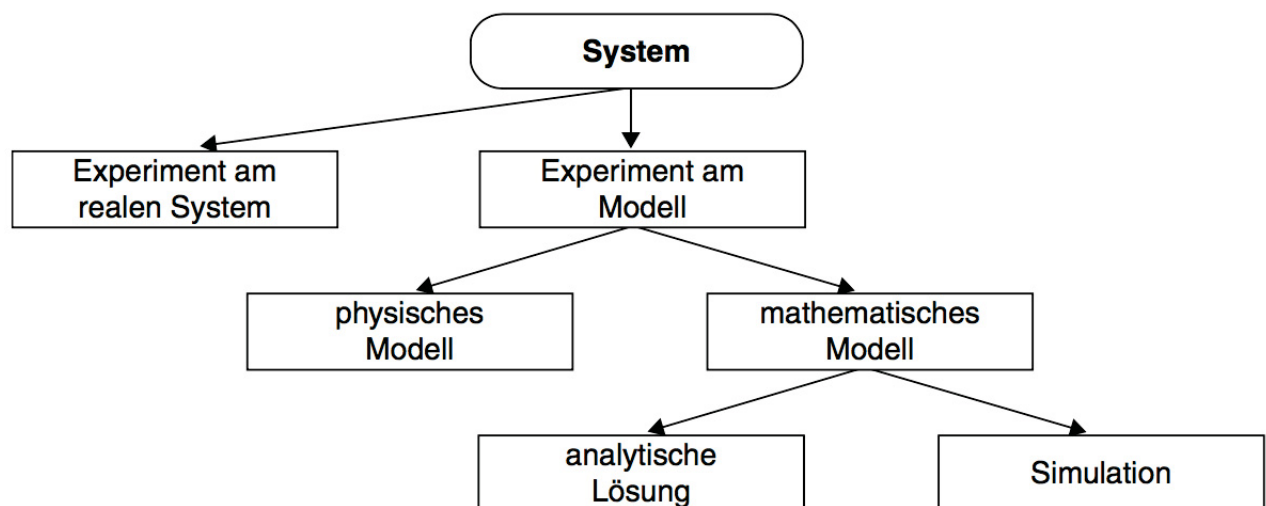


Abbildung 3.1: Methoden zur Untersuchung eines Systems⁴¹

Die Simulation ist ein wichtiges Instrumentarium zur Ergänzung von theoretischer Analyse und Experimenten, allerdings kann sie diese nicht zur Gänze ersetzen.⁴²

³⁹Vgl. Eley 2012, S.3 ff.

⁴⁰Vgl. Daniluk und Chisu 2010, S.147.

⁴¹Quelle: Daniluk und Chisu 2010, S.148.

⁴²Vgl. Bungartz et al. 2009, S.2.

3.1.2 Weitere Definitionen

Für das Verständnis der folgenden Kapitel sind außerdem zwei weitere Begriffe von Bedeutung, welche im Anschluss erläutert werden.

Simulationslauf:

Ein Simulationslauf, oft auch Experiment genannt, ist ein Durchlauf der Simulation. Das heißt, das Modell wird einmalig ausgeführt, was in einer definierten Zeitspanne, dem sogenannten Simulationszeitraum, passiert. Sollten im Modell zufallsabhängige Größen verwendet werden, müssen mehrere Simulationsläufe durchgeführt werden, um die Ergebnisse zu verifizieren und statistische Abweichungen zu eliminieren. Durch diese Replikation werden zuverlässige Ergebnisse erhalten.⁴³

Simulationszeit und Rechenzeit:

Die Simulationszeit ist eine Abstraktion, die von der Simulation verwendet wird, um die im realen System voranschreitende Zeit abzubilden. Sie ist nicht zu verwechseln mit der Rechenzeit, im Englischen auch als „wallclock time“ bezeichnet. Diese bezeichnet die Zeit, welche benötigt wird, um ein System über einen bestimmten Simulationszeitraum zu analysieren. Bei Computersimulationen ist dies die Zeit, die während der Ausführung eines Simulationsprogramms vergeht.⁴⁴

3.1.3 Simulationspipeline

Die Definition der Simulation aus Kapitel 3.1 ist sehr allgemein gehalten. Im Wesentlichen bezeichnet Simulation laut Bungartz et al.⁴⁵ „...den Gesamtkomplex der Vorberechnung oder des Nachstellens eines bestimmten Szenarios.“ Für diese Arbeit ist vor allem die Computersimulation relevant, weshalb im Folgenden speziell darauf eingegangen wird. Bungartz et al.⁴⁶ beschreiben Simulationen im weiteren Sinne auch als „... virtuelle Experimente am Computer.“

⁴³Vgl. Rabe et al. 2008, S.12.

⁴⁴Vgl. Fujimoto 2000, S.27; Eley 2012, S.4.

⁴⁵Bungartz et al. 2009, S.1.

⁴⁶Bungartz et al. 2009, S.1.

Eine Simulation besteht aus mehreren Schritten, ist im Normalfall aber kein linearer Prozess. Vielmehr werden in sogenannten Feedback-Schleifen die einzelnen Schritte mehrmals durchlaufen. Die Gesamtheit dieser Schritte wird als Simulationspipeline, dargestellt in Abbildung 3.2, bezeichnet.⁴⁷

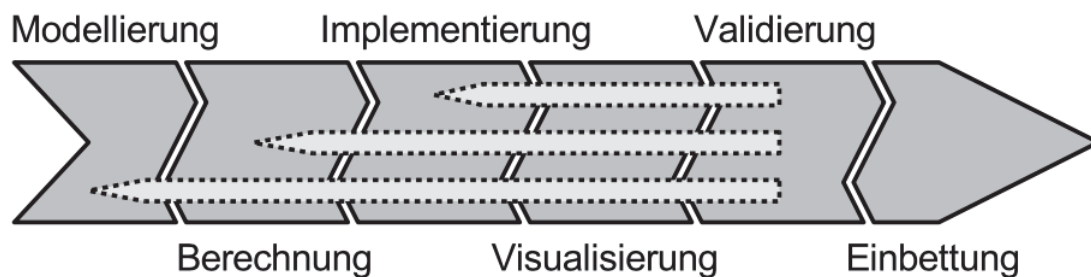


Abbildung 3.2: Simulationspipeline⁴⁸

Im Folgenden werden die einzelnen Schritte der Simulationspipeline genauer beleuchtet:

- **Modellierung:**

Wie bereits in Kapitel 3.1.1 beschrieben, ist die Voraussetzung für eine Simulation ein zuvor erstelltes Modell des realen Systems. Dieses bildet die Grundlage für die nachfolgenden Berechnungen.⁴⁹

Vor der Erstellung des Modells ist es wesentlich, den Detaillierungsgrad sorgfältig zu wählen. Ein gewisser Detaillierungsgrad ist notwendig, um verwertbare Ergebnisse zu erhalten. Bei zu hoher Detaillierung steigt aber der Zeit- und Kostenaufwand der Modellbildung sowie die Rechenzeit für die anschließend durchgeführten Experimente bzw. Simulationsläufe. Die genaue Definition eines Simulationslaufs findet sich in Kapitel 3.1.2. Da im Normalfall nur begrenzte Zeit zur Verfügung steht, führt dies dazu, dass insgesamt weniger Experimente durchgeführt werden können, was wiederum die Aussagekraft des Ergebnisses schwächt.⁵⁰

⁴⁷Vgl. Bungartz et al. 2009, S.2 ff.

⁴⁸Quelle: Bungartz et al. 2009, S.3.

⁴⁹Vgl. Bungartz et al. 2009, S.3.

⁵⁰Vgl. Kuhn und Rabe 1998, S.3.

- **Berechnung bzw. Simulation im engeren Sinne:**

„Das Modell wird geeignet aufbereitet (z. B. diskretisiert), um es auf dem Rechner behandeln zu können, und zur Lösung dieses aufbereiteten Modells sind effiziente Algorithmen zu ermitteln.“⁵¹

Erst wenn das Modell in mathematisch-logischer Form vorliegt, kann es im nächsten Schritt von einem Computerprogramm implementiert werden.⁵²

- **Implementierung oder Software-Entwicklung:**

In diesem Schritt müssen die Berechnungsalgorithmen aus dem vorherigen Schritt implementiert werden. Dies soll möglichst effizient in Bezug auf Rechenzeit, Speicherkomplexität und Parallelisierbarkeit erfolgen. Die dazu verwendeten Computerprogramme werden in diesem Zusammenhang auch als Simulationswerkzeuge betitelt.⁵³

- **Visualisierung oder Datenexploration:**

Nach Durchführung eines Simulationslaufes müssen die erhaltenen Daten interpretiert und die relevante Information aus ihnen extrahiert werden.⁵⁴

- **Validierung:**

Hier kommen die vorhin erwähnten Feedback-Schleifen ins Spiel, denn die erlangten Ergebnisse müssen überprüft werden. Deshalb werden verschiedene Modelle, Algorithmen und Codes miteinander abgeglichen. Wenn sich Fehler finden, so muss die Pipeline ab jenem Schritt, in dem die entsprechende Fehlerquelle liegt, aufs Neue durchlaufen werden, was in Abbildung 3.2 durch die rückwärts zeigenden Pfeile deutlich gemacht wird.⁵⁵

Die Validierung untersucht Aspekte der Eignung der verwendeten Verfahren. Sie versucht z.B. die Frage, „Ist es das richtige Modell?“, zu beantworten. Der Begriff fällt oft gemeinsam mit Verifikation, welche auf Korrektheit überprüft, also beispielsweise eine Antwort auf die Frage, „Ist das Modell richtig?“, sucht. Tatsächlich sind die beiden Termini schwer zu trennen und werden teilweise auch

⁵¹Bungartz et al. 2009, S.3.

⁵²Vgl. Eley 2012, S.4.

⁵³Vgl. Bungartz et al. 2009, S.3; Eley 2012, S.4.

⁵⁴Vgl. Bungartz et al. 2009, S.3.

⁵⁵Vgl. Bungartz et al. 2009, S.3.

synonym verwendet.⁵⁶

▪ **Einbettung:**

„Simulationen finden in einem Kontext statt – ein Entwicklungs- oder Produktionsprozess beispielsweise – und sollen in diesen integriert werden. Dies erfordert Schnittstellendefinition, ein vernünftiges Software Engineering, einfache Testumgebungen, etc.“⁵⁷

Jeder dieser sechs Schritte bringt unterschiedliche Herausforderungen mit sich. Allerdings ist es nicht sinnvoll, die einzelnen Schritte von verschiedenen Experten bearbeiten zu lassen, um sie anschließend zu einem Gesamtkonzept zusammenzufügen. Denn alle Schritte sind eng miteinander verbunden und bei der Bearbeitung des einen Schritts müssen auch die anderen bereits im Auge behalten werden. Vor allem die ersten beiden Schritte, „Modellbildung“ und „Simulation im engeren Sinne“, sind von großer Bedeutung.⁵⁸

3.1.4 Simulationsstudie

In diesem Kapitel wird zum besseren Verständnis noch eine weitere Darstellungsmöglichkeit des Simulationsprozesses vorgestellt. Im Unterschied zur Simulationspipeline, die trotz Feedback-Schleifen als linearer Prozess ausgeführt ist, verwendet die Simulationsstudie eine kreisförmige Darstellung, welche Abbildung 3.3 zeigt. Dies soll verdeutlichen, dass Ergebnisse aus der Simulationsstudie sofort eingesetzt werden können, um das reale System zu verbessern. Im Anschluss kann der Prozess von vorne beginnen, indem ein neues Modell des nun verbesserten Systems erstellt wird.⁵⁹

Noch bevor die Modellierung des Systems erfolgen kann, muss die Simulationswürdigkeit des jeweiligen Systems überprüft werden. Das bedeutet, es muss festgestellt werden, ob nicht eine andere Lösungsmethode (vgl. Abbildung 3.1) besser zur Lösung dieses Problems geeignet wäre. Außerdem müssen finanzieller und zeitlicher Aufwand

⁵⁶Vgl. Wenzel, Weiß et al. 2007, S.33.

⁵⁷Bungartz et al. 2009, S.3 ff.

⁵⁸Vgl. Bungartz et al. 2009, S.4.

⁵⁹Vgl. Eley 2012, S.4.

der Simulationsstudie ihrem Nutzen gegenübergestellt werden, um ihre Durchführung zu rechtfertigen.⁶⁰

Fällt die Entscheidung zur Durchführung einer Simulationsstudie, so müssen anschließend das Gesamtziel sowie damit verbundene Teilziele spezifiziert werden. Danach muss eine Datenbasis mit allen für die Simulation erforderlichen Daten aufgebaut werden. Dies stellt bei realen Systemen in der Regel kein großes Problem dar. Soll aber eine Simulationsstudie für ein geplantes und noch nicht existentes System durchgeführt werden, gestaltet sich die Datenbeschaffung meist schwieriger, da hier noch keine Erfahrungswerte existieren und teilweise auch konkrete Daten fehlen. In diesem Fall müssen Daten ähnlicher, bereits bestehender Systeme zu Hilfe gezogen werden.⁶¹

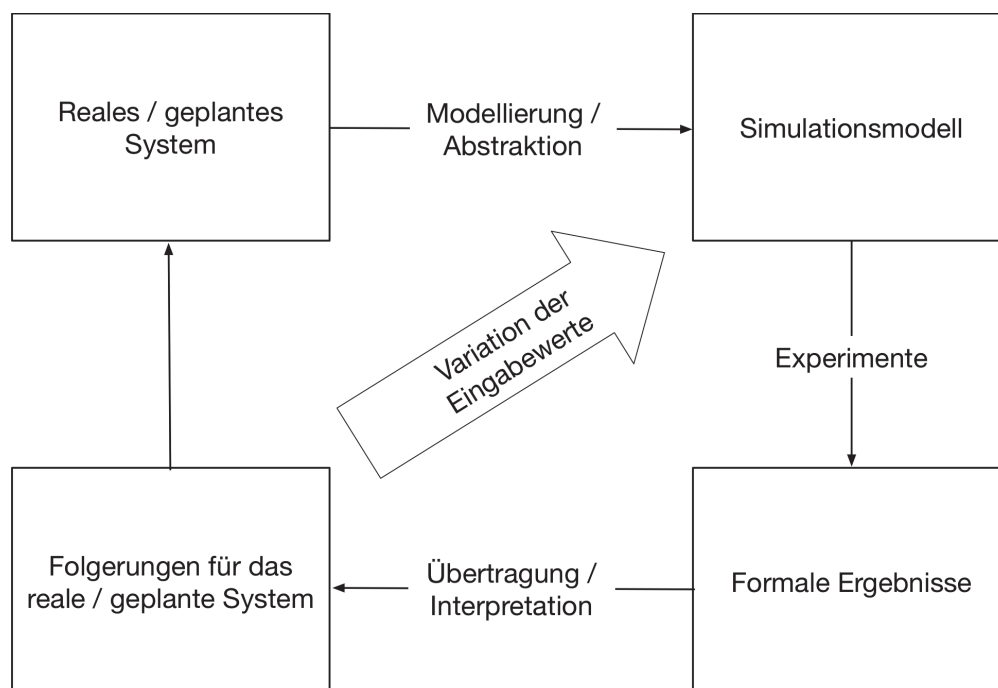


Abbildung 3.3: Durchführung einer Simulationsstudie⁶²

⁶⁰Vgl. Daniluk und Chisu 2010, S.149.

⁶¹Vgl. Daniluk und Chisu 2010, S.150.

⁶²Vgl. Hrdliczka et al. 1997, S.3.

Nach dem Abschluss dieser Vorbereitungsarbeiten startet der bereits in Kapitel 3.1.3 beschriebene Modellierungsprozess sowie die Implementierung des Modells. Anschließend folgt die gezielte Untersuchung des Modellverhaltens anhand mehrerer Simulationläufe. Hierfür müssen zuerst sinnvolle Versuchsreihen erstellt werden, die festlegen, wie einzelne Parameter bei unterschiedlichen Simulationsläufen gewählt werden, um brauchbare Ergebnisse für das zuvor definierte Ziel zu erhalten. Um aus den gewonnenen Daten Rückschlüsse für das reale System ziehen zu können, müssen selbige zuvor aufbereitet werden. Dies kann mittels unterschiedlicher Darstellungsformen, wie zum Beispiel Animationen oder Statistiken, erfolgen. Es ist außerdem sinnvoll, diverse Kennzahlen zu definieren, um unterschiedliche Systeme miteinander vergleichen zu können.⁶³

Eine Simulationsstudie kann aus unterschiedlichen Gründen erstellt werden. Einerseits kann das Verhalten eines Systems analysiert werden, andererseits kann aber auch die Entwicklung eines Systems gemäß bestimmten Vorgaben das Ziel sein. Mögliche Anwendungsgebiete von Simulationen werden im nächsten Kapitel angeführt.⁶⁴

3.2 Anwendungsgebiete einer Simulation

Simulationen werden bereits in vielen unterschiedlichen Bereichen wie Technik, Naturwissenschaften und Gesellschaftswissenschaften verwendet. Daniluk und Chisu⁶⁵ definieren typische Einsatzgebiete von Simulationen. Sie finden z.B. Verwendung in Situationen, in denen:

- Neuland beschritten wird
- die Grenzen analytischer Methoden erreicht sind
- komplexe Wirkzusammenhänge die menschliche Vorstellungskraft überfordern
- das Experimentieren am realen Modell nicht möglich bzw. zu kostenintensiv ist
- das zeitliche Ablaufverhalten einer Anlage untersucht werden soll

⁶³Vgl. Daniluk und Chisu 2010, S.150 ff.

⁶⁴Vgl. Eley 2012, S.4.

⁶⁵Daniluk und Chisu 2010, S.148.

3.2.1 Simulation in der Logistik

Im Bereich der Logistik findet die Simulation ein großes Anwendungsgebiet. Sowohl in der Intralogistik als auch im Rahmen der Supply Chain kommen Simulationsstudien zum Einsatz. Die folgenden Kapitel behandeln aber vor allem den Bereich der Intralogistik.⁶⁶

Insbesondere die Planungsphase bietet sich für den Einsatz von Simulationen an, da hier Probleme mit hoher Komplexität auftreten. Weil der Einsatz in der Planungsphase sehr naheliegend ist, wird hier bereits am längsten von Simulationen Gebrauch gemacht. In der heutigen Zeit werden sie jedoch im gesamten Lebenszyklus von logistischen Systemen verwendet und finden sowohl in ihrer Planung als auch in der Realisierung und im Betrieb zahlreiche Einsatzmöglichkeiten. Im Folgenden wird genauer auf diese drei Anwendungsbereiche eingegangen.⁶⁷

3.2.1.1 Planung

Die Planung eines Systems ist besonders essentiell, da hier bereits mögliche spätere Probleme vermieden werden können. Fragestellungen, die in der Planungsphase auftreten, sind, wie bereits erwähnt, oft hoch komplex, da sehr viele unterschiedliche Einflüsse zu berücksichtigen sind. Mittels analytischer Methoden können sie nicht oder nur sehr schwer beantwortet werden. Zu ermittelnde Faktoren sind beispielsweise Engpässe und Schwachstellen des Systems oder die Höhe und Streuung der Durchlaufzeiten von Aufträgen. Auch das Anlaufverhalten eines Systems und sein Verhalten bei Störungen oder Spitzenbelastungen sind von zentraler Bedeutung. Ein weiteres Beispiel ist die Beantwortung der Frage, ob eine bestimmte Kombination von Aufträgen innerhalb eines vorgegebenen Zeitraums erfüllt werden kann. All diese Fragestellungen bereits vor der Errichtung des realen Systems beantworten zu können, bringt enorme Vorteile mit sich.⁶⁸

⁶⁶Vgl. Wenzel und Peter 2017, S.14 ff.

⁶⁷Vgl. Eley 2012, S.5; Daniluk und Chisu 2010, S.148.

⁶⁸Vgl. Daniluk und Chisu 2010, S.149.

Kuhn und Rabe⁶⁹ liefern eine Auflistung all der positiven Aspekte, die durch den Einsatz von Simulation in der Planungsphase erzielt werden können:

- Absicherung der Planung durch das Sammeln von „Erfahrungen“ an dem noch nicht realisierten System
- Überprüfung der Funktionalität der Anlage und dadurch ggf. die Möglichkeit zur rechtzeitigen Korrektur
- Korrekte Dimensionierung der Anlage, z. B. durch Optimierung von Puffergrößen und Lagerbeständen
- Einsparung oder Vereinfachung von Systemelementen, die ohne Simulation häufig „auf der sicheren Seite“ ausgelegt und dadurch unnötig überdimensioniert werden
- Sensitivitätsanalyse, die Aussagen zur Reaktion des Systems auf Schwankungen in den Eingangsgrößen (z.B. Störungen, Produktspektrum) liefert
- Überprüfung des Pflichtenheftes, indem dessen Inhalte in das Simulationsmodell übertragen und die resultierenden Ergebnisse gemessen werden

3.2.1.2 Realisierung

Auf die Planungsphase folgt die Realisierungsphase des Systems. Die Anlagen wurden im Normalfall bereits bestellt, aber die Steuerung muss noch detailliert werden. Der Aufbau der Anlagen bietet eine weitere Chance mögliche Fehler frühzeitig zu erkennen und außerdem muss das Personal eingeschult werden. Es ist von Vorteil, wenn die Inbetriebnahme möglichst schnell vonstattengeht, weil dadurch schneller der volle Durchsatz und damit die Gewinnzone erreicht werden kann. Simulationen können bei der Verkürzung der Inbetriebnahmezeit helfen und bieten damit ein hohes Kosteneinsparungspotential. Wenn bereits in der Planungsphase eine Simulationsstudie erstellt wurde, so kann diese weiter verfeinert werden.⁷⁰

⁶⁹Kuhn und Rabe 1998, S.7 ff.

⁷⁰Vgl. Kuhn und Rabe 1998, S.8; Haußner et al. 2010, S.170.

Laut Kuhn und Rabe⁷¹ bringen Simulationen in dieser Phase Vorteile durch:

- Bestimmung von Notfallstrategien beim Ausfall einzelner Anlagenkomponenten, sodass entsprechende Erfahrungen nicht erst im Betrieb gewonnen werden müssen
- Darstellung der Funktionalität und der Zusammenhänge
- Das ineinandergreifende Entwickeln und Testen von Steuerungssoftware
- Mitarbeiterschulung am simulierten System, die dadurch wesentlich früher beginnen kann

3.2.1.3 Betrieb

In der Betriebsphase des Systems können Simulationen unterstützend eingesetzt werden. Sie helfen dem Betriebspersonal, Entscheidungen z.B. bei Störungen zu treffen. Im Gegensatz zur Planungsphase ist der betrachtete Zeitraum hier aber im Normalfall relativ kurz. Daher kann keine allgemeine Lösung für Störungen gefunden werden, sondern diese müssen bei ihrem Auftreten einzeln abgehandelt werden und für die gegebene Situation muss kurzfristig eine passende Strategie gewählt werden. Weiters können Simulationen auch in der Mitarbeiterschulung eingesetzt werden.⁷²

Folgende Einsatzfelder gibt es nach Kuhn und Rabe⁷³ für Simulationen in der Betriebsphase:

- Die vorausschauende Untersuchung von Strategien, zwischen denen kurzfristig gewechselt werden kann
- Die Überprüfung einer Einplanung der Aufträge auf die verfügbaren Anlagen
- Die Reaktion auf Störfälle

3.3 Vor- und Nachteile von Simulationen

Als Abschluss dieses Kapitels folgt eine Zusammenfassung wichtiger Vor- und Nachteile, die durch den Einsatz einer Simulation auftreten können. Diese beziehen sich vor

⁷¹Kuhn und Rabe 1998, S.8.

⁷²Vgl. Kuhn und Rabe 1998, S.8; Daniluk und Chisu 2010, S.148.

⁷³Kuhn und Rabe 1998, S.8.

allem auf den Vergleich zwischen Experimenten mittels Simulation und Experimenten am realen System. Dabei ist zu beachten, dass die Möglichkeit, Letztere durchzuführen, nicht immer besteht.

Zuerst sollen die Nachteile betrachtet werden, welche außerdem das Einsatzgebiet von Simulationen eingrenzen. Denn diese können nicht als Allheilmittel verstanden werden, das einfache zu erlangende Antworten auf komplexe Fragestellung liefert. Systeme mit großem Umfang und vielen unterschiedlichen Einflüssen führen sowohl bei der Modellerstellung als auch bei der Ergebnisinterpretation zu einem entsprechend hohen Aufwand. Ein essentieller Bestandteil der Simulation ist außerdem die Datenbeschaffung. Denn die Qualität der Ergebnisse ist durch die Qualität des Inputs beschränkt. Falsche oder unvollständige Eingangsdaten führen demnach dazu, dass keine vertrauenswürdigen Schlussfolgerungen auf das reale System gezogen werden können.⁷⁴

Außerdem ist die Durchführung einer Simulation verglichen mit analytischen Methoden relativ aufwendig. Daher sollte sie nur zum Einsatz kommen, wenn der Einsatz anderer, schnellerer Methoden nicht möglich ist.⁷⁵

Unter den richtigen Bedingungen kann der Einsatz von Simulation jedoch große Vorteile mit sich bringen. Die Durchführung von Experimenten ist in weitaus geringerer Zeit möglich, als dies im realen System der Fall ist. Daher können weit mehr Durchläufe ausgeführt werden, was wiederum die statistische Genauigkeit der Ergebnisse erhöht bzw. die Beantwortung bestimmter Fragestellungen überhaupt erst ermöglicht. Auch wenn für große, komplexe Systeme ein höherer Aufwand betrieben werden muss, so können mittels Simulation doch Probleme gelöst werden, die für analytische Methoden unüberwindbar wären. Des Weiteren kann die Anschaulichkeit von Simulationen komplizierte Zusammenhänge bildhaft darstellen, was zu einem besseren Verständnis des Gesamtsystems beiträgt. Alle erwähnten Vor- und Nachteile sind zur besseren Übersicht in Tabelle 3.1 erneut angeführt.⁷⁶

⁷⁴Vgl. Daniluk und Chisu 2010, S.151 ff.

⁷⁵Vgl. Kuhn und Rabe 1998, S.7.

⁷⁶Vgl. Daniluk und Chisu 2010, S.151.

Tabelle 3.1: Vor- und Nachteile von Simulationen⁷⁷

Vorteile	Nachteile
<ul style="list-style-type: none"> ▪ Die Durchführung von Experimenten ist in viel geringerer Zeit möglich, als im realen System. ▪ Durch zusätzliche Durchläufe kann die statistische Genauigkeit erhöht werden. ▪ Bestimmte Fragestellungen sind für analytische Methoden nicht beantwortbar und können erst durch den Einsatz einer Simulation gelöst werden. ▪ Mittels einer Simulation können komplizierte Zusammenhänge anschaulich dargestellt werden. 	<ul style="list-style-type: none"> ▪ Simulation ist eine vergleichsweise sehr aufwendige Methode. ▪ Umfangreiche Systeme mit vielen Einflussfaktoren führen zu einem hohen Aufwand bei der Modellerstellung und der Ergebnisinterpretation. ▪ Es kann mit großen Anstrengungen verbunden sein, die nötigen Daten zur Durchführung der Simulation zu beschaffen. ▪ Falsche oder unvollständige Eingangsdaten führen zu unrichtigen oder nicht vertrauenswürdigen Ergebnissen.

Obige Ausführungen erklären allgemeine Vor- und Nachteile von Simulationen und machen deutlich, dass ein sinnvoller Einsatz sehr stark von den vorliegenden Umständen abhängt. Wie Simulation zur Durchführung der gewünschten Tests gewinnbringend eingesetzt werden kann, soll in Kapitel 5 dargelegt werden. Durch die Erläuterung der erstellten Software können alle Vorteile des Testsystems klar aufgezeigt werden.

⁷⁷Vgl. Kuhn und Rabe 1998, S.7; Daniluk und Chisu 2010, S.151 ff.

4 Auswahl eines geeigneten Testsystems

In Kapitel 2.3 wurde erklärt, dass vom Host-System Kommissionier- und Replenishmentaufträge ans WCS weitergeleitet werden. Die Auftragsdaten werden in Form von Nachrichten ans WCS übermittelt und anschließend in dessen Produktionstabellen eingetragen. Um die Performance des WCS zu testen, müssen ebensolche Nachrichten generiert werden. Daher ist es von wesentlicher Bedeutung, den Aufbau der Nachrichten zu kennen, weshalb dieser im folgenden Abschnitt erläutert wird.

4.1 Nachrichtenstruktur

Eine Nachricht kann in vier ineinander verschachtelte Stufen untergliedert werden, wie in Abbildung 4.1 dargestellt ist. Pro Nachricht wird immer genau ein Event-Eintrag erzeugt, der Auskunft über die Art der Nachricht gibt. In einem Event werden Informationen für einen oder mehrere Transportaufträge übermittelt. Die Daten für einen Auftrag finden sich im Container-Eintrag.

Einträge für Function Point (FP) und Working Step (WS) kann es ebenso mehrere geben. Ein FP-Eintrag definiert ein Ziel eines Auftrags bzw. eine Arbeitsstation und der WS-Eintrag bezeichnet einen Arbeitsschritt auf dieser Station. Ein jeder dieser vier Einträge besteht aus mehreren Feldern, die je nach Nachrichtentyp unterschiedlich befüllt werden, wobei nicht in jeder Nachricht alle zur Verfügung stehenden Felder tatsächlich mit Werten belegt sind.

Im Normalfall bezieht sich eine Nachricht genau auf einen Auftrag, das heißt, es gibt auch nur einen Container-Block. Um die Gesamtzahl der Nachrichten zu verringern, können aber in manchen Fällen mehrere Aufträge in einer Nachricht zusammengefasst

werden, was mehrere Container-Blöcke in einem Event zur Folge hat. Die einzelnen Stufen und wichtige Felder dieser Stufen werden in den folgenden Abschnitten genauer beschrieben.⁷⁸

EVENT number 1

CONTAINER number 1 (of EVENT number 1)

FP number 1 (of CONTAINER number 1)

WS number 1 (of FP number 1)

...

WS number n (of FP number 1)

...

FP number m (of CONTAINER number 1)

WS number 1 (of FP number m)

...

WS number n (of FP number m)

...

Abbildung 4.1: Aufbau einer Nachricht vom Host ans WCS⁷⁹

4.1.1 Event

Der Event-Eintrag ist sozusagen die Wurzel für alle weiteren Einträge und muss in jeder Nachricht vorhanden sein. Jedes Event wird mit einer EVENT_ID versehen. Weitere wichtige Felder in diesem Eintrag sind ACTION, SUB_ACTION, IDENTIFIER, COMMAND und PRIORITY. Die Kombination der beiden Felder ACTION und SUB_ACTION kann als Typ der Nachricht verstanden werden. Die unterschiedlichen Arten von Nachrichten und die daraus resultierenden Testfälle werden in Kapitel 4.2 genauer beschrieben.

⁷⁸Interne Dokumente

⁷⁹Interne Dokumente

IDENTIFIER und COMMAND definieren die Nachricht genauer. Eine hohe PRIORITY wird für Nachrichten mit wichtigen Daten verwendet, die unverzüglich gesendet werden sollen. Alle hier erwähnten Felder finden sich gemeinsam mit einer Beschreibung und einem Beispiel in Tabelle 4.1.⁸⁰

Tabelle 4.1: Felder des Event-Eintrags⁸¹

Feldname	Beschreibung	Inhalt bzw. Beispiel
EVENT_ID	Identifikationsnummer des Events	ganzzahliger Wert
ACTION	Aktion, die von der Nachricht ausgelöst wird	ADD
SUB_ACTION	Genauere Spezifizierung der Aktion	CONTAINER
IDENTIFIER	Identifikationshilfe	vordefinierte Zeichenkette
COMMAND	Befehl (Identifikationshilfe)	ADD
PRIORITY	Priorität	ganzzahliger Wert

4.1.2 Container

Auf der zweithöchsten Ebene findet sich der Container-Eintrag, welcher genau einen Transportauftrag beschreibt. Allerdings bezeichnet dieser Eintrag auch den physischen Container, da im konkreten Fall ein Auftrag immer aus genau einem Container besteht. Alle Felder dieses Eintrags sind in Tabelle 4.2 aufgelistet und werden im Folgenden näher beschrieben.

Das Feld EVENT_ID verweist auf das Event, das diesen Eintrag beinhaltet. Charakterisiert wird der Transportauftrag durch die WPO_NO, einer für das ganze System einzigartigen Auftragsnummer. WPO steht hier für Warehouse Processing Order. Außerdem wird jeder Container mit einer Nummer, der CNT_NO sowie mit der LICENCEPLATE

⁸⁰Interne Dokumente

⁸¹Interne Dokumente

(Barcode) versehen. Auch der aktuelle Status des Auftrags wird im Feld STATUS übermittelt. Er kann Werte wie z.B. „übergeben“, „beendet“ oder „widerrufen“ annehmen. Die WP_TYPE_ID charakterisiert den Typ des Auftrag, die CONT_TYPE_ID die Art des Containers. Besteht die Notwendigkeit, Behälter ans Distributionslager zurückzusenden, so wird die RETURN_REFERENCE benötigt. Sie wird verwendet, um sich auf den ursprünglichen Transport zu beziehen.⁸²

Tabelle 4.2: Felder des Container-Eintrags⁸³

Feldname	Beschreibung	Inhalt bzw. Beispiel
EVENT_ID	Identifikationsnummer des Events	ganzzahliger Wert
WPO_NO	Auftragsnummer	ganzzahliger Wert
CNT_NO	Containernummer	ganzzahliger Wert
LICENCEPLATE	Barcode	ganzzahliger Wert
STATUS	Status des Containers	finished (beendet)
WP_TYPE_ID	Typ des Auftrags	vordefinierte Zeichenkette
CONT_TYPE_ID	Art des Containers	vordefinierte Zeichenkette
RETURN_REFERENCE	Referenznummer bei Rücksendungen	ganzzahliger Wert
CUTOFF_TIME	Cut-off-Zeit	Datum und Uhrzeit
CUTOFF_UPD_ALLOWED	Gibt an, ob die Cut-off-Zeit verändert werden kann	Y (Ja) oder N (Nein)
SHIP_VIA	Lieferunternehmen	DHL
REPACK_SIZE	Kartongröße	vordefinierte Zeichenkette
DEST_CNTRY	Zielland	Austria
VAS_CODE	Anweisung zum Verpacken	Liste bestimmter Codes

⁸²Interne Dokumente

⁸³Interne Dokumente

Die Cut-off-Zeiten, welche angeben, wann der Behälter das Lager spätestens verlassen haben muss, werden im Feld CUTOFF_TIME festgelegt und ob diese im Nachhinein verändert werden dürfen, wird in CUTOFF_UPD_ALLOWED angeführt. Weiters werden das verwendete Lieferunternehmen (SHIP_VIA), die Kartongröße (REPACK_SIZE) und das Zielland (DEST_CNTRY) definiert. Im Feld VAS_CODE finden sich außerdem Anweisungen, wie der Auftrag verpackt werden soll.⁸⁴

4.1.3 FP (Function Point)

Jeder Auftrag hat mindestens ein Ziel, welches im FP-Eintrag festgelegt wird. FPs bezeichnen Arbeitsstationen, wie z.B. Quellen oder Senken. In diesem Eintrag finden sich weit weniger Felder, als im zuvor beschriebenen Container-Eintrag. Die EVENT_ID verweist wiederum auf das zugehörige Event, während WPO_NO und CNT_NO auf den jeweiligen Container-Eintrag referenzieren. Der Name des Ziels bzw. der Station wird im Feld MASTER_FP_NAME definiert. Im Feld STATUS wird ebenso auf den aktuellen Status verwiesen und für einige Arten von Nachrichten ist außerdem die Angabe der LICENCEPLATE notwendig. Eine Auflistung aller Felder ist in Tabelle 4.3 dargestellt.⁸⁵

Tabelle 4.3: Felder des FP-Eintrags⁸⁶

Feldname	Beschreibung	Inhalt bzw. Beispiel
EVENT_ID	Identifikationsnummer des Events	ganzzahliger Wert
WPO_NO	Auftragsnummer	ganzzahliger Wert
CNT_NO	Containernummer	ganzzahliger Wert
MASTER_FP_NAME	Name der Zielstation	OSR
STATUS	Status	tansferred (übergeben)
LICENCEPLATE	Container Barcode (notwendig für spezielle Nachrichten)	ganzzahliger Wert

⁸⁴Interne Dokumente

⁸⁵Interne Dokumente

⁸⁶Interne Dokumente

4.1.4 WS (Working Step)

Auf jeder der im Auftrag involvierten Stationen gibt es im Normalfall mehrere Arbeitsschritte. Ein solcher Arbeitsschritt wird im WS-Eintrag beschrieben. Tatsächlich bezieht sich dieser Eintrag im konkreten Fall auf eine spezifische Lagereinheit bzw. Stock Keeping Unit (SKU). Wichtige Attribute dieses Artikels werden in den folgenden Feldern, die in Tabelle 4.4 aufgezählt werden, beschrieben.⁸⁷

Tabelle 4.4: Felder des WS-Eintrags⁸⁸

Feldname	Beschreibung	Inhalt bzw. Beispiel
EVENT_ID	Identifikationsnummer des Events	ganzzahliger Wert
WPO_NO	Auftragsnummer	ganzzahliger Wert
CNT_NO	Containernummer	ganzzahliger Wert
MASTER_FP_NAME	Name der Zielstation	OSR
STATUS	Status	tansferred (übergeben)
WS_TYPE	Art der Bearbeitung	vordefinierte Zeichenkette
QTY	Anzahl verschiedener Artikel	ganzzahliger Wert
ITEM_NO	Artikelnummer	Kombination aus Zahlen und Buchstaben
CNTRY_ORIGIN	Herkunftsland	Austria
VAS_CODE	Hinweise zum Verpacken	Liste bestimmter Codes
HOST_LINE_NO	Referenznummer zum ERP-System	ganzzahliger Wert

Auch hier finden sich wieder EVENT_ID, WPO_NO und CNT_NO, die auf Event und Container-Eintrag referenzieren. Der MASTER_FP_NAME nimmt Bezug auf die zuge-

⁸⁷Interne Dokumente

⁸⁸Interne Dokumente

hörige Arbeitsstation. Der aktuelle Status wird auch in diesem Eintrag im Feld STATUS beschrieben. Im Feld WS_TYPE wird die Art der Bearbeitung definiert, also es wird übermittelt, ob SKU's hinzugefügt, entfernt oder nur gezählt werden. Die Anzahl der zu bearbeitenden SKU's wird in QTY angegeben und ITEM_NO enthält die Artikelnummer. Außerdem wird das Herkunftsland im Feld CNTRY_ORIGIN übermittelt und VAS_CODE weist weitere Hinweise zum Verpacken auf. Die HOST_LINE_NO bezieht sich auf die Zeilennummer im ERP-System.⁸⁹

4.2 Testsituationen und -mechanismen

In Kapitel 4.1 wurde die Struktur der vom Host ans WCS übermittelten Nachrichten erklärt. Um ein passendes Testsystem auswählen zu können, muss zuvor noch genau festgelegt werden, welche Situationen konkret getestet werden sollen und welche Arten von Testfällen auftreten können.

Ein wichtiges Szenario ist das Hinzufügen eines Wareneingangs, also die Bekanntgabe, dass Waren geschickt werden und eingelagert werden sollen. Es kann allerdings auch vorkommen, dass ein solcher Auftrag im Nachhinein widerrufen werden muss. Auch hierfür muss eine Möglichkeit geschaffen werden, dies zu testen. Ebenso muss natürlich ein neuer Warenausgang, also ein neuer Kommissionierauftrag, hinzugefügt werden können. Wie beim Wareneingang muss es die Möglichkeit geben, diesen wieder zu stornieren. Außerdem treten Fälle auf, in denen gewisse Daten eines Kommissionierauftrags verändert werden müssen, was ebenfalls getestet werden soll.

Weiters soll das Senden leerer Behälter ans WCS übermittelt werden können. Auch Rücksendungen verschiedener Artikel ans Distributionslager sind aus verschiedenen Gründen notwendig und müssen getestet werden, ebenso wie der Widerruf solcher Rücksendungen. Außerdem ist das Zuweisen von und das Bestätigen bereits zugewiesener Versanddrampen notwendig.

Alle zur Verfügung stehenden Artikel sind in einer Datei aufgelistet, auf welche das WCS zugreift. Es soll die Ankündigung einer neuen Datei, also eines neuen Artikelsortiments, getestet werden. Ferner ist das Sperren und auch das Freigeben einzelner

⁸⁹Interne Dokumente

Artikel im Lagerbestand notwendig.⁹⁰

Aus all diesen Situationen lassen sich drei Testfälle herausfiltern. Einerseits müssen Einträge unterschiedlicher Art hinzugefügt werden können, andererseits ist aber auch das Entfernen zuvor generierter Daten notwendig. Die dritte Möglichkeit ist die Veränderung bereits vorhandener Einträge. Diese verschiedenen Testfälle können leicht durch die unterschiedlichen Befehle im Feld ACTION des Event-Eintrags erkannt werden. In den folgenden Abschnitten sollen sie genauer beleuchtet und die Unterschiede hervorgehoben werden.

4.2.1 Hinzufügen eines Eintrags - ADD

Nachrichten dieses Typs sind durch den Befehl ADD im Feld ACTION zu erkennen. Sie sollen neue Einträge in den Produktionstabellen des WCS generieren. Normalerweise ist bei solchen Nachrichten eine Befüllung vieler Felder notwendig, da eine große Menge an Informationen übermittelt werden muss.

In den meisten Fällen werden neue Aufträge, also Container-Einträge, hinzugefügt. Für diese muss jeweils eine neue Auftragsnummer generiert werden. Es gibt allerdings auch Anwendungsfälle, in denen nur neue Ziele in Form von FP-Einträgen hinzugefügt werden sollen, wie etwa bei der Zuweisung einer neuen Versandrampe. Auch das Artikelsortiment kann durch solche Nachrichten erweitert werden.

4.2.2 Löschen eines Eintrags - DEL

Diese Art von Nachricht kann am Befehl DEL im Feld ACTION erkannt werden. Im Gegensatz zum eben beschriebenen Fall dient sie dazu, bereits vorhandene Einträge aus den Produktionstabellen des WCS zu entfernen. Hierzu muss natürlich auf den zu löschenden Eintrag Bezug genommen werden.

Diese Nachricht wird hauptsächlich dazu verwendet, gesendete Warenein- und -ausgänge zu widerrufen. Die übermittelte Auftragsnummer muss hierbei der Nummer des zu entfernenden Auftrags entsprechen. Da es hier nur darum geht, den Eintrag aus den Produktionstabellen zu löschen, müssen nicht viele Informationen übermittelt werden und die meisten Felder bleiben bei dieser Art von Nachricht leer.

⁹⁰Interne Dokumente

4.2.3 Modifizieren eines Eintrags - MOD

Außer dem Hinzufügen und dem Entfernen von Einträgen muss außerdem das nachträgliche Modifizieren eines Eintrags möglich sein. Der Befehl MOD im Feld Action kennzeichnet Nachrichten dieses Typs.

Auch hier muss natürlich der Bezug zu dem Eintrag, der verändert werden soll, hergestellt werden. Bei Aufträgen geschieht dies über die Auftragsnummer. Ein Anwendungsfall hierfür findet sich, wenn ein Kommissionierauftrag im Nachhinein modifiziert werden soll. Auch das Sperren und Freigeben einzelner Artikel wird mit dieser Art von Nachrichten ermöglicht. Der Bezug zum Originaleintrag wird hierbei durch die Artikelnummer geschaffen.

4.3 Anforderungen an ein Testsystem

Dieser Abschnitt soll erläutern, welche Anforderungen das Testsystem erfüllen muss, um die unterschiedlichen Testsituationen aus Kapitel 4.2 zu ermöglichen. Die im Anschluss ausgeführten Vorgaben sind notwendig, um Testdaten zu erstellen, die den Realfall möglichst gut nachbilden.

Es muss auf jeden Fall die Möglichkeit bestehen, einzelne Nachrichten mit genau vorgegebenen Parametern zu senden, um die korrekte Arbeitsweise bestimmter Funktionen zu gewährleisten. Andererseits soll aber auch die Übermittlung großer Mengen von Aufträgen in kurzer Zeit durchführbar sein, damit das System bei hoher Auslastung getestet werden kann. Die Auswahl der einzelnen Parameter soll hierbei nach festgelegten Regeln erfolgen, die für jeden Parameter unterschiedlich ausfallen können. Diese Anweisungen sind für die Erstellung der Nachrichten vorgegeben. Für einzelne Felder werden beispielsweise Verteilungen in Form von Listen mit zugehörigen Wahrscheinlichkeiten vorgegeben. Die Gesamtheit der übermittelten Werte muss diesen Vorgaben entsprechen. Bei numerischen Werten kann auch ein Zahlenspektrum vorgegeben sein, aus dem die Felder mit zufälligen Werten befüllt werden sollen.

Werden die Nachrichten händisch angefertigt, so stellt es kein großes Problem dar, diese Vorgaben zu erfüllen, da ja alle Werte einzeln ausgewählt werden können. Soll

aber ein System geschaffen werden, das Nachrichten automatisch erstellt, so muss dafür Sorge getragen werden, dass auch so diese Vorgaben eingehalten werden können. Für die Auswahl der übermittelten Artikel ist beispielsweise eine Datenbank mit allen Beständen vorhanden. Bei der Generierung der Testdaten sollen aus dieser die Artikel für die einzelnen Aufträge zufällig ausgewählt werden. Außerdem findet sich in dieser Datenbank der aktuelle Lagerbestand aller Waren. Bei der Auswahl eines Artikels für einen Kommissionierauftrag muss beachtet werden, ob der Vorrat noch ausreicht, die gewünschte Anzahl zu liefern, da dies im Realfall natürlich eine logische Bedingung zur erfolgreichen Erfüllung eines Auftrages ist. Beinhaltet ein Auftrag mehr als einen WS-Eintrag, muss weiters darauf geachtet werden, dass nicht für mehrere dieser Einträge derselbe Artikel ausgewählt wird. Um Verwirrung zu vermeiden, darf es für jeden Artikel pro Auftrag nur einen Eintrag geben. Bei zusätzlichen Mengen kann einfach die Anzahl erhöht werden.⁹¹

Ein geeignetes Testsystem muss also dazu in der Lage sein, genaue Vorgaben zu erfüllen. Außerdem muss es unterschiedliche Arten von Tests bewältigen können. Im nächsten Abschnitt werden diese kurz beschrieben.

4.3.1 Arten von Tests

Tests, die mit dem Testsystem durchgeführt werden sollen, lassen sich in zwei verschiedene Typen unterteilen. Mit jeder dieser Testarten werden unterschiedliche Ziele verfolgt und jede für sich ist demnach unabdingbar. In den folgenden Abschnitten sollen die beiden Typen erklärt werden. Im Besonderen wird auf die Unterschiede zwischen den Arten und den Zweck der Ausführung hingewiesen.

4.3.1.1 Funktionale Tests

Funktionale Tests sind notwendig, um zu prüfen, ob das System bestimmte Funktionen und Aufgaben, die von ihm verlangt werden, auch erfüllen kann. So werden hier einzelne Test mit besonders ausgefallenen Bedingungen durchgeführt. Es geht also nicht um den Standardfall, in dem der Durchschnittsartikel im Durchschnittspaket verschickt wird, sondern Ausreißer von der Norm sollen getestet werden. Im Fall von Artikeln

⁹¹Interne Dokumente

bzw. Paketen wird hier zum Beispiel ein besonders großes oder ein sehr schweres Paket ausgewählt. Damit kann gezeigt werden, dass das System auch Extremfälle bewältigen kann.

Weitere Beispiele für solche Ausnahmesituationen, die bei funktionalen Tests zum Einsatz kommen, sind die Vorgabe einer sehr kurzen Zeit zur Erfüllung des Auftrags sowie die Auswahl einer ganz bestimmten Route im Lagersystem.⁹²

4.3.1.2 Massentests

Im Gegensatz zu den oben beschriebenen funktionalen Tests, überprüfen Massentests nicht bestimmte einzelne Szenarien, sondern testen, wie der Name schon sagt, mit einer großen Anzahl von Aufträgen. Sie sind notwendig, um die Kapazitäten des Systems auszutesten. Die Auswahl der Artikel wird auch hier vorgegeben, allerdings ist es nicht von Bedeutung einen einzelnen bestimmten Artikel zu wählen. Vielmehr wird ein Prozentsatz vorgegeben nach dem Artikel mit bestimmten Kriterien selektiert werden, um gewisse Anforderungen zu erfüllen. Massentests händisch zu generieren stellt einen kaum bewältigbaren Aufwand dar.⁹³

4.3.2 Arten von Nachrichten

In diesem Kapitel wurde bereits erklärt, welche Aufgaben das Testsystem bewältigen muss und welche Arten von Tests mit ihm durchgeführt werden sollen. Nun folgt eine genaue Auflistung all der Nachrichten, die es ans WCS übermitteln soll, um die Bereitschaft des Systems zu prüfen.

4.3.2.1 Ankündigung eines Wareneingangs

Das Host-System teilt dem WCS mit dieser Nachricht mit, dass ein Behälter mit verschiedenen Artikeln geliefert wird und eingelagert werden soll. Sobald der Behälter im Lager eintrifft, wird der Barcode gescannt und eine Bestätigungsnachricht wird zurück an den Host gesandt.

⁹²Interne Dokumente

⁹³Interne Dokumente

Der Container-Eintrag dieser Nachricht enthält die in Kapitel 4.1.2 beschriebenen Inhalte, wobei natürlich Felder, wie CUTOFF_TIME und SHIP_VIA, die nur für Warenausgänge notwendig sind, leer bleiben. FP-Eintrag gibt es nur einen, denn dieser Auftrag hat nur ein Ziel und zwar das OSR (Order Storage Retrieval), da die Artikel ja eingelagert werden sollen.

WS-Einträge kann es beliebig viele geben, je nachdem wie viele SKU's ans Lager geliefert werden. Ein WS-Eintrag enthält die Artikelnummer sowie die jeweils übermittelte Anzahl. Weiters beinhaltet er das Herkunftsland und die Information, dass der Artikel ein- und nicht ausgelagert werden soll.⁹⁴

4.3.2.2 Widerrufung eines Wareneingangs

Diese Nachricht bietet die Möglichkeit, einen bereits angekündigten Wareneingang zu stornieren. Mittels der Angabe von WPO_NO, LICENCEPLATE und RETURN_REFERENCE kann der entsprechende Auftrag identifiziert werden und das WCS kann entsprechend reagieren. Da es sich um einen Widerruf handelt, kann in dieser Nachricht kein Ziel angegeben werden und somit ist kein FP-Eintrag notwendig. Auch WS-Einträge sind nicht erforderlich, weil die in diesem Auftrag enthaltenen SKU's nicht erneut angegeben werden müssen.⁹⁵

4.3.2.3 Ankündigung eines leeren Behälters

Wenn vom Distributionslager leere Behälter geschickt werden, so muss auch dies dem WCS angekündigt werden, was mittels dieser Nachricht geschieht. Im Container-Eintrag sind vor allem die WPO_NO des Auftrags sowie die LICENCEPLATE des Containers wichtig. Ein FP-Eintrag ist vorhanden, allerdings ist der MASTER_FP_NAME mit einem vordefinierten „leeren“ Eintrag und keinem tatsächlichen Ziel befüllt. Da es sich um einen Behälter ohne Inhalt handelt, sind keine WS-Einträge für etwaige Artikel nötig.⁹⁶

⁹⁴Interne Dokumente

⁹⁵Interne Dokumente

⁹⁶Interne Dokumente

4.3.2.4 Ankündigung eines neuen Kommissionierauftrags

Mit dieser Nachricht werden dem WCS vom Host alle Informationen zu einem neuen Kommissionierauftrag mitgeteilt, so dass das WCS diesen Auftrag bearbeiten kann. Die meisten dieser Informationen finden sich, wie in Kapitel 4.1.2 beschrieben, im Container-Eintrag. Per Definition wird im MASTER_FP_NAME des FP-Eintrags wie beim Wareneingang auch hier das OSR als Ziel übergeben. Je nachdem wie viele spezifische Lagereinheiten der Auftrag enthält, werden entsprechend viele WS-Einträge übermittelt. Diese enthalten vor allem die Artikelnummer und die Anzahl des jeweiligen Artikels sowie die Informationen, dass es sich um eine Auslagerung handelt.⁹⁷

4.3.2.5 Stornierung eines Kommissionierauftrags

Ebenso wie für einen Wareneingang gibt es auch für einen Kommissionierauftrag die Möglichkeit, diesen zu widerrufen. Um dies zu kommunizieren, wird diese Nachricht vom Host ans WCS geschickt. Der entsprechende Auftrag wird durch Übermittlung von WPO_NO und CNT_NO identifiziert. Auch hier sind weder FP- noch WS-Eintrag notwendig.⁹⁸

4.3.2.6 Modifizierung eines Kommissionierauftrags

Im Gegensatz zu Wareneingängen ist es möglich, einen Kommissionierauftrag im Nachhinein zu verändern. Dies geschieht mit Hilfe dieser Nachricht. Sie wird verwendet, um die Cut-off-Zeiten bzw. das verwendete Lieferunternehmen zu aktualisieren.

Im Normalfall wird diese, wie auch alle bisher beschriebenen Nachrichten, durch den Host initiiert. Allerdings kann hier ein Szenario auftreten, in welchem das Host-System auf eine Nachricht des WCS reagieren muss. Dies tritt ein, wenn vom WCS errechnete Cut-off-Zeiten überschritten werden. In diesem Fall schickt das WCS eine Nachricht, in der es neue Cut-off-Zeiten anfordert, woraufhin der Host mit der Modifizierung eines Kommissionierauftrags antwortet.

Der Bezug zum zu verändernden Auftrag wird wieder über WPO_NO und CNT_NO hergestellt und die neuen Informationen werden in CUTOFF_TIME und SHIP_VIA

⁹⁷Interne Dokumente

⁹⁸Interne Dokumente

übermittelt. FP- und WS-Einträge des Originalauftrags müssen nicht wiederholt werden.⁹⁹

4.3.2.7 Anforderung einer Rücklieferung

Diese Nachricht fordert die Rücklieferung von Artikeln ans Distributionslager an, was aus verschiedenen Gründen notwendig sein kann. WPO_NO und RETURN_REFERENCE werden angegeben, um den Bezug herzustellen. Außerdem hat dieser Auftrag zwei Ziele, daher beinhaltet er auch zwei FP-Einträge. Das erste Ziel ist das OSR, da die Artikel hier entnommen werden. Die einzelnen WS-Einträge umfassen die Artikelnummer, die Artikelanzahl sowie die Information, dass ausgelagert wird.

Als zweites Ziel wird das Distributionslager angeführt, an das die Artikel geliefert werden. Hierfür sind allerdings keine weiteren WS-Einträge nötig, da sich die Informationen bereits im ersten FP-Eintrag finden.¹⁰⁰

4.3.2.8 Anforderung der Rücklieferung bei Saisonende

Im Grunde ist diese Nachricht genauso aufgebaut, wie jene aus Abschnitt 4.3.2.7. Das Feld COMMAND im Event-Eintrag wird unterschiedlich befüllt, um die beiden unterscheiden zu können. Außerdem wird bei Saisonende der gesamte Bestand retourniert, weswegen keine Angabe der Artikelanzahl notwendig ist.¹⁰¹

4.3.2.9 Widerrufung einer Rücklieferung

Auch eine Rücklieferung kann wieder storniert werden. Diese Nachricht wird verwendet, um eine der beiden Nachrichten aus den obigen Abschnitten 4.3.2.7 und 4.3.2.8 zu widerrufen. Über WPO_NO und RETURN_REFERENCE wird der entsprechende Auftrag identifiziert. Es gibt keine Einträge für FP und WS.¹⁰²

⁹⁹Interne Dokumente

¹⁰⁰Interne Dokumente

¹⁰¹Interne Dokumente

¹⁰²Interne Dokumente

4.3.2.10 Hinzufügen einer Versandrampe

Diese Nachricht stellt auf zwei unterschiedliche Arten einen Spezialfall dar. Erstens wird hier nur eine Zielstation und kein gesamter Auftrag hinzugefügt, wodurch kein Container-Eintrag notwendig ist. Der übliche Aufbau der Nachrichten aus Abbildung 4.1 wird hier also nicht vollständig eingehalten, da der Event-Eintrag direkt den FP-Eintrag beinhaltet. In diesem wird der Name der Zielstation, also die Versandrampe, übermittelt. Außerdem wird hier das Feld LICENCEPLATE benötigt, um auf den Barcode des Containers zu verweisen, für den die Versandrampe hinzugefügt werden soll. Zweitens wird diese Nachricht nicht vom Host initiiert, sondern als Antwort auf eine Nachricht des WCS geschickt. Das bedeutet, es muss dafür gesorgt werden, dass sie automatisch ausgelöst wird.¹⁰³

4.3.2.11 Bestätigung einer Versandrampe

Auch diese Nachricht wird als Antwort auf eine Nachricht des WCS geschickt. Sobald ein Behälter einen gewissen Scanner passiert, wird dem Host mitgeteilt, dass alles zur Verladung bereitsteht. Mittels dieser Nachricht erhält das WCS die Bestätigung für eine bestimmte Versandrampe. Der Aufbau ist nahezu analog zur Nachricht aus Abschnitt 4.3.2.10. Die Unterscheidung der beiden Nachrichten erfolgt wiederum durch unterschiedliche Befüllung des Feldes COMMAND im Event-Eintrag.¹⁰⁴

4.3.2.12 Ankündigung neuer Artikeldaten

Mit dieser Nachricht wird dem WCS mitgeteilt, dass eine neue Datei mit Artikeldaten vorliegt. In dieser befinden sich die Informationen, welche unterschiedlichen SKU's zur Verfügung stehen. Weder Container noch FP- oder WS-Einträge sind für diese Nachricht nötig. Im Event-Eintrag wird der Name der neuen Datei übermittelt, so dass das WCS weiß, worauf zugegriffen werden soll.¹⁰⁵

¹⁰³Interne Dokumente

¹⁰⁴Interne Dokumente

¹⁰⁵Interne Dokumente

4.3.2.13 Sperren/Entsperren eines Artikels

Aus verschiedenen Gründen kann es notwendig sein, einzelne Artikel im Lagerbestand zu sperren, so dass diese für keine weiteren Aufträge verwendet werden können. Dies geschieht mittels dieser Nachricht. Die dafür nötigen Informationen können nicht in Container-, FP- oder WS-Einträgen übermittelt werden, weshalb für diese Nachricht direkt nach dem Event-Eintrag ein eigener STOCK-Eintrag eingeführt wird. In diesem wird dem WCS die Artikelnummer sowie die Information, ob dieser Artikel gesperrt oder wieder entsperrt werden soll, mitgeteilt. Es können beliebig viele STOCK-Einträge in einer Nachricht übermittelt werden, je nachdem für wie viele Artikel sich der Sperrstatus ändern soll.¹⁰⁶

4.4 Mögliche Lösungsansätze

Nachdem in Kapitel 4.3 die Anforderungen an ein Testsystem dargelegt wurden, sollen nun verschiedene Ansätze aufgezeigt werden, um diese Anforderungen zu erfüllen. Die nachfolgenden Abschnitte beschreiben unterschiedliche Möglichkeiten, wie die nötigen Tests durchgeführt werden können.

4.4.1 Manuelle Datenerstellung

Die einfachste Option ist, die Nachrichten, die ans WCS gesendet werden sollen, per Hand zu erstellen. Auf diese Weise können genau die gewünschten Testdaten erzeugt werden. Im Folgenden wird zwischen zwei Möglichkeiten der Durchführung unterschieden.

4.4.1.1 Erstellung mittels Textfile

Hierbei wird ein Textfile kreiert, das den Aufbau der Nachricht mit allen notwendigen Informationen enthält. Dies stellt eine gute Möglichkeit dar, bestimmte funktionale Tests durchzuführen. Das Erstellen jeder einzelnen Nachricht ist allerdings mit einem recht hohen Zeitaufwand verbunden. Für Einzelfälle kann diese Art zu testen also durchaus

¹⁰⁶Interne Dokumente

verwendet werden, sobald aber eine größere Anzahl von Tests notwendig ist, stößt diese Methode an ihre Grenzen. Massentests durchzuführen ist mit ihr überhaupt nicht möglich, da jede einzelne Nachricht eigens erstellt werden müsste, was einen nicht bewältigbaren zeitlichen Aufwand darstellen würde.

Zu Beginn eines Projekts kann diese Methodik allerdings gut eingesetzt werden, da man ohne großen vorherigen Aufwand erste Nachrichten erstellen kann. Konkrete Fälle können rasch getestet werden und erste Antworten werden schnell erhalten.

4.4.1.2 Erstellung mittels MS Excel

Eine weitere Möglichkeit, die nötigen Testdaten zu erhalten, ist die Verwendung von Microsoft Excel. Bei Verwendung dieser Methodik werden Tabellen mit Einträgen für die Produktionstabellen des WCS mittels MS Excel erzeugt und anschließend in diese eingepflegt.

Das Erstellen der Nachrichten erfolgt auf diese Weise bereits weit schneller, als mit der in Abschnitt 4.4.1.1 beschriebenen Möglichkeit. Gibt es viele ähnliche Einträge, die nur geringe Unterschiede aufweisen, kann durch Kopieren und anschließendes Ändern von Kleinigkeiten eine große Zeitersparnis erfolgen. Es können auch Vorlagen erstellt werden, die das Verfassen der Nachrichten beschleunigen. Außerdem ist MS Excel ein Programm, das weithin bekannt ist und von Benutzern bedient werden kann, ohne dass eine längere Einschulung notwendig ist.

Trotzdem benötigt auch diese Methode bei einer hohen Anzahl von Nachrichten einen großen zeitlichen Aufwand. Massentests mit sehr vielen Aufträgen können auch auf diese Weise nicht durchgeführt werden, da auch hier jede Nachricht händisch erstellt werden muss.

4.4.2 Automatische Datengenerierung - Einsatz einer Simulation

Um dem händischen Erstellen einzelner Nachrichten zu entgehen, muss ein Weg geschaffen werden, diese automatisch zu generieren. Dies kann mittels einer Simulation bewerkstelligt werden. Auch ist nur durch eine Simulation das automatische Antworten auf Nachrichten des WCS an den Host möglich.

Allerdings muss zur Errichtung einer Simulation ein größerer Aufwand betrieben werden, bevor mit dem Testen begonnen werden kann, als bei den bereits beschriebenen Methoden. Ist diese jedoch erstellt, ist ein schnelleres und im besten Fall auch benutzerfreundlicheres Testen möglich. Kapitel 3 lieferte bereits einen allgemeinen Einblick in das Thema Simulation. Die genaue Vorgehensweise zur Erstellung der Host-Simulation sowie die Durchführung der Tests werden in Kapitel 5 erläutert.

5 Durchführung der Tests mittels Simulation

Kapitel 3 lieferte den theoretischen Hintergrund zum Thema Simulation und in Kapitel 4 wurden alle notwendigen Anforderungen an ein Testsystem dargelegt. Dieses Kapitel soll diese beiden Punkte nun verknüpfen und zeigen, wie die Durchführung der Tests mit der Simulation als Testsystem abläuft. Zu Beginn wird die konkrete Erstellung der Host-Simulation erläutert.

5.1 Implementierung der Host-Simulation

Als erster Schritt musste eine geeignete Programmiersprache zur Implementierung ausgewählt werden. Die Wahl fiel auf PL/SQL (Procedural Language extensions to the Structured Query Language), da sie, wie der Name bereits sagt, die Abfragesprache SQL zu einer prozeduralen Programmiersprache erweitert. SQL wird verwendet, um Abfragen und Aktualisierung an Datenbanken vorzunehmen. Durch PL/SQL wurden einige Einschränkungen von SQL aufgehoben, um zusätzliche Möglichkeiten in der Arbeit mit Datenbanken zu bieten und trotzdem alle SQL-Befehle auf einfache Weise verwenden zu können.¹⁰⁷

Da die Produktionstabellen des WCS befüllt werden sollen und sich somit für die Simulation eine Arbeit mit Datenbanken anbietet, war die Verwendung von PL/SQL eine naheliegende Lösung.

¹⁰⁷Feuerstein und Pribyl 2009, S.3.

5.1.1 Erstellung der nötigen Tabellen

Schon in Kapitel 4.3 wurde erklärt, dass die Erstellung der Nachrichten bestimmten Vorgaben folgen muss. Einige dieser Vorgaben sind für alle Nachrichten eines Typs gleich, andere können vom Benutzer bestimmt werden. So soll der Anwender beispielsweise für Wareneingänge eine Liste möglicher Herkunftsländer mit zugehörigen Wahrscheinlichkeiten für die Auswahl eines bestimmten Landes angeben können. Die Anweisungen des Benutzers werden in Tabellen gespeichert, aus denen später die Informationen extrahiert werden, welche benötigt werden, um den Vorgaben entsprechende Nachrichten zu erstellen.

Daher mussten zu Beginn die Parameter definiert werden, welche vom Anwender festgelegt werden können. Anschließend wurden die Parametertabellen, in denen die Vorgaben des Benutzers gespeichert werden, erstellt.

Die generierten Event-, Container-, FP- und WS-Einträge werden nicht sofort in die Produktionstabellen des WCS integriert. Nach der Erzeugung werden sie in einer Datenbank zwischengespeichert, um anschließend ans WCS übermittelt werden zu können. Der nächste Schritt war also die Erstellung der Nachrichtentabellen zur Zwischenspeicherung der Einträge.

Außerdem war die Erstellung von Tabellen für die möglichen Artikel für einen Wareneingang sowie für den aktuellen Lagerbestand notwendig. Die nötigen Informationen dafür, wie Artikelnummer und -beschreibung, mussten vom WCS importiert und laufend synchronisiert werden. Für die Durchführung einzelner Szenarien musste Sorge getragen werden, dass nur Artikel mit den gewünschten Kriterien ausgewählt werden können. Vorgaben des Benutzer beziehen sich hierbei auf die Artikelnummer und/oder -beschreibung.

Weiters wurden temporäre Tabellen für die aktuelle Artikelauswahl erzeugt. Bei der Erstellung der Nachrichten wurden diese unter Berücksichtigung des Umstandes, ob eine Überprüfung des Lagerbestandes notwendig ist und welche Artikel in einer Nachricht bereits ausgewählt wurden, befüllt, um die derzeit verfügbaren Artikel anzuzeigen.

5.1.2 Implementierung der Prozeduren

Anschließend folgte die konkrete Implementierung zur Erzeugung der einzelnen Nachrichtentypen aus Kapitel 4.3.2. Vom Benutzer werden hierfür Szenarien erstellt, welche Informationen enthalten, um zwischen den verschiedenen Arten von Nachrichten differenzieren zu können und dementsprechende Einträge zu generieren. Neben den genauen Vorgaben für die Befüllung der einzelnen Felder werden außerdem noch die gewünschte Anzahl an Container-Einträgen pro Event sowie an WS-Einträgen pro FP vom Benutzer angegeben. Die Anzahl der FP-Einträge ist im Normalfall durch den Nachrichtentyp vorgegeben.

Die Implementierung wurde nacheinander für die verschiedenen Arten von Nachrichten vorgenommen. Dafür mussten unterschiedliche Prozeduren in PL/SQL erstellt werden. Um duplizierten Code zu vermeiden, wurden gleiche Operationen, die in mehrere Nachrichtentypen vorhanden sind, zusammengefasst. Begonnen wurde mit Nachrichten zum Hinzufügen von Einträgen in den Produktionstabellen des WCS, also solchen mit dem Befehl ADD im Feld ACTION. Eine Besonderheit stellten hierbei jene Nachrichtentypen dar, die für das Hinzufügen oder Bestätigen von Versanddrampen zuständig sind, da diese keine Container-Einträge beinhalten und die entsprechenden Prozeduren dadurch anders aufgebaut werden mussten.

Für die verschiedenen Nummern, wie EVENT_ID, WPO_NO oder LICENCEPLATE mussten Sequenzen generiert werden, welche zur Erzeugung der jeweiligen IDs herangezogen wurden. Außerdem wurden einige Hilfsfunktionen geschaffen, um die Werte zur Befüllung der Nachrichtentabellen nach den Vorgaben in den Parametertabellen zu erzeugen. Ein einfaches Beispiel hierfür ist die zufällige Auswahl eines der Werte aus einer Liste unter Berücksichtigung der gegebenen Wahrscheinlichkeiten.

Nach den Prozeduren zum Hinzufügen von Einträgen in den Produktionstabellen des WCS folgte die Erstellung der Prozeduren zur Entfernung dieser Einträge. Auch hierfür musste entsprechender Input für die Nachrichtentabellen erstellt werden, allerdings mit dem Befehl DEL im Feld ACTION. Im Vergleich zu ersteren, sind diese Prozeduren weit weniger umfangreich. Vom Benutzer werden hier nur die EVENT_ID sowie die WPO_NO und die CNT_NO zur Identifizierung eines bestimmten Containers überge-

ben. Weitere Informationen zu diesem Auftrag können aus dem bereits vorhandenen Eintrag in den Nachrichtentabellen entnommen werden. Außerdem sind für diese Art von Nachrichten keine FP- und WS-Einträge notwendig. Bei Übermittlung der Nachrichten ans WCS werden die zugehörigen Einträge aus dessen Produktionstabellen entfernt.

Im Anschluss wurden noch die Prozeduren zum Verändern von Einträgen erstellt. Genauer gesagt gibt es nur eine Nachricht, die zum Modifizieren von Einträgen verwendet wird und den Aufbau aus Abbildung 4.1 aufweist. Diese wurde in Abschnitt 4.3.2.6 beschrieben und wird dazu verwendet, die Daten eines Auftrags zu aktualisieren.

Wie beim Löschen von Einträgen werden auch hierfür `EVENT_ID`, `WPO_NO` und `CNT_NO` vom Benutzer übergeben, um auf den entsprechenden Auftrag referenzieren zu können. Weiters müssen natürlich auch die Werte für die zu verändernden Felder angegeben werden. In den Nachrichtentabellen werden wiederum Einträge erzeugt, in diesem Fall mit dem Befehl `MOD` im Feld `ACTION` gekennzeichnet. Nicht zu verändernde Informationen werden abermals aus den vorhandenen Einträgen in den Nachrichtentabellen entnommen, zu aktualisierende Felder werden mit den übergebenen Werten beschrieben. Die Übermittlung dieser Nachrichten ans WCS bewirkt die Abänderung der entsprechenden Einträge in den Produktionstabellen.

Jene Nachrichten, die nicht der allgemeinen Struktur folgen, mussten gesondert betrachtet werden. Um auf eine Datei mit neuen Artikeldaten hinzuweisen, musste, wie bereits erwähnt, nur ein Event-Eintrag erstellt werden. In diesem wird dem WCS der Dateiname mitgeteilt.

Um einen bestimmten Artikel im Lagerbestand zu sperren oder wieder freizugeben, muss der Benutzer die Artikelnummer übergeben sowie die Information, ob der Artikel verfügbar ist oder nicht. Hierfür musste zusätzlich eine eigene Nachrichtentabelle `STOCK` erstellt werden, in der diese Informationen zur anschließenden Übermittlung ans WCS gespeichert werden.

Wie in Kapitel 4.3.2 dargelegt wurde, sollen einige Nachrichten automatisch geschickt werden, wenn eine bestimmte Nachricht vom WCS einlangt. Um dies zu bewerkstel-

ligen, mussten sogenannte Trigger erstellt werden, die das Senden von Nachrichten auslösen, sobald bestimmte Bedingungen vorliegen.

Für den Benutzer sollte es möglich sein, einzelne Szenarien zu erstellen. Ein Szenario bezeichnet hierbei die konkrete Ausführung einer Art von Nachricht mit bestimmten Werten in den Parametertabellen. Daher mussten abschließend die Prozeduren zum Erstellen, Verändern und Löschen dieser Szenarien implementiert werden. Außerdem waren weitere Prozeduren zum Hinzufügen, Verändern und Entfernen einzelner Parameter aus diesen Szenarien notwendig.

5.1.3 Erstellung der Benutzeroberfläche

Wie bereits erwähnt, sollte für den Anwender die Möglichkeit gegeben sein, einzelne Szenarien mit Parametern zu erstellen. Deshalb musste eine Oberfläche geschaffen werden, um dem Benutzer das Eintragen der gewünschten Werte zu ermöglichen und ihm die Ergebnisse auf übersichtliche Weise anzuzeigen.

Zu diesem Zweck war bereits eine Umgebung vorhanden, welche auch bei anderen Projekten im Einsatz ist. Damit konnte eine Oberfläche zur einfachen Bedienung geschaffen werden, in der all die unterschiedlichen Prozeduren zur Erzeugung der Nachrichten eingepflegt werden konnten.

In dieser Umgebung mussten zunächst Anzeigen für die Parameter-, die Nachrichten- und die Artikeltabellen geschaffen werden. Hierfür war es notwendig, die entsprechenden Tabellen aus der Datenbank zu importieren. Außerdem mussten die einzelnen Felder mit Metadaten versehen werden, die z.B. die Spaltenüberschriften angeben. Die Anzeige der einzelnen Tabellen für sich war aber nicht ausreichend, da für den Benutzer daraus nicht sofort ersichtlich ist, welcher Container-Eintrag z.B. zu welchem Event-Eintrag gehört. Daher war es erforderlich, Verknüpfungen zwischen den einzelnen Tabellen zu schaffen und damit übersichtliche Ansichten zu erstellen.

Die Hauptansicht zur Anzeige der Nachrichtentabellen wurde so aufgebaut, wie es der Nachrichtenaufbau aus Abbildung 4.1 nahelegt. Sie beinhaltet demnach die vier Tabellen für Event-, Container-, FP- und WS-Einträge. Über die entsprechenden Inhalte in

EVENT_ID, WPO_NO, CNT_NO, und MASTER_FP_NAME wurde die Verknüpfung zwischen den einzelnen Tabellen hergestellt. Für die Nachrichtenarten ohne Container-Eintrag musste eine weitere Anzeige erstellt werden, in der vom Event- direkt auf den FP-Eintrag verwiesen wurde. Außerdem war eine Anzeige mit einer Verknüpfung von der Event- auf die Artikeltabelle notwendig.

Analog dazu wurde eine Anzeige mit den vom Benutzer erstellten Szenarien und einer Verknüpfung auf die einzelnen Parameter pro Szenario erstellt. Eine genauere Beschreibung der einzelnen Anzeigen sowie eine genaue Erklärung zur Bedienung folgt in Kapitel 5.2.

Der nächste Schritt beinhaltete das Inkludieren der Prozeduren in die zuvor erstellten Anzeigen. Einerseits waren dies jene zum Erstellen, Verändern und Löschen der Szenarien und deren Parameter, andererseits die Prozeduren zur Erzeugung der Nachrichten. Auch hierfür mussten wiederum Metadaten erstellt werden.

Die Prozeduren wurden an den entsprechenden Stellen eingepflegt und für die Übergabe der einzelnen Parameter mussten Richtlinien erstellt werden. Für den Großteil war die Eingabe einer beliebige Zeichenkette ausreichend, für andere musste eine Liste erzeugt werden, aus der ein Wert ausgewählt werden konnte und wieder andere wurden automatisch aus einer Tabelle übernommen. Wie die Verwendung der unterschiedlichen Prozeduren abläuft, wird im folgenden Kapitel genau erläutert.

5.2 Bedienung der Simulation

Um die Simulation erfolgreich bedienen zu können, muss zuerst die richtige Anzeige ausgewählt werden. Daher wird in diesem Kapitel erklärt, welche Anzeige welche Funktion hat und wo die unterschiedlichen Prozeduren genau eingebaut wurden. Dies erfolgt anhand der einzelnen Schritte, die zur Bedienung der Simulation notwendig sind.

5.2.1 Erstellung eines Szenarios

Der erste Schritt zur Verwendung der Simulation ist das Kreieren eines Szenarios. Dies ist in der Szenariokonfiguration möglich, in der die einzelnen Szenarien mit den jeweils

zugehörigen Parametern dargestellt sind. Abbildung 5.1 zeigt ihren Aufbau.

Im oberen Bereich der Abbildung finden sich die verschiedenen bereits erstellten Szenarien. Wichtig sind hierbei vor allem die ID und der Name des Szenarios. Bei Aufruf dieser Anzeige kann auch ein Filter angewandt werden, um z.B. nur Szenarien auszuwählen, die eine bestimmte Formulierung in ihrem Namen enthalten.

Der untere Bereich der Abbildung 5.1 beinhaltet die einzelnen Parameter des ausgewählten Szenarios. Diese sind in vier Kategorien, welche in unterschiedlichen Tabs dargestellt werden, unterteilt. Im ersten Tab „Scenario Parameters“ finden sich allgemeine Angaben wie der Typ des Szenarios sowie die Anzahl der Aufträge. Die weiteren Tabs enthalten die Angaben zu den jeweiligen Container-, FP- und WS-Einträgen. Da ein WS-Eintrag sich in diesem Projekt immer auf eine SKU bezieht, wurde der zugehörige Tab mit „SKU Parameters“ betitelt.

Die „Scenario ID“ bezieht sich immer auf das zugehörige Szenario. Für die „Parameter ID“ wird vom Benutzer ein gewisses Vorwissen verlangt, da hier eine bestimmte vordefinierte Zeichenkette einzutragen ist, die in den Prozeduren zur Erstellung der Nachrichten benötigt wird. Der „Parameter Value Type“ gibt an, was in diesem Eintrag übergeben wird. Möglichkeiten dafür sind:

- ein einzelner boolescher Wert
- eine Zeichenkette
- eine Zahlenspanne von einem niedrigen bis zu einem höheren Wert
- eine Liste von Werten
- eine zu einer Liste zugehörige Wahrscheinlichkeitsverteilung
- eine Artikelnummer oder ein Teil einer Artikelnummer

Der tatsächliche Wert wird in der Spalte „Parameter Value Low“ angegeben. „Parameter Value High“ wird nur bei einer Zahlenspanne benötigt.

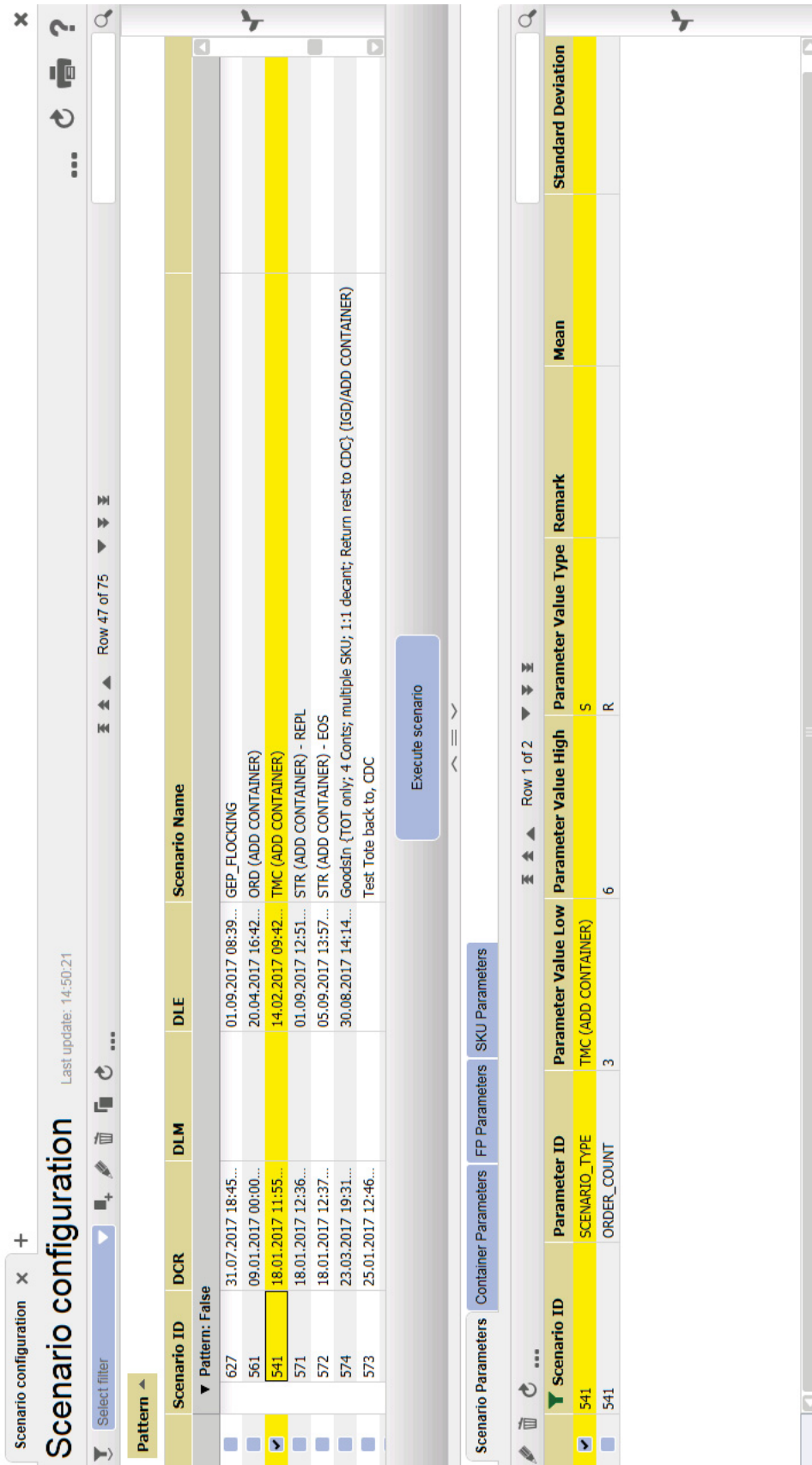


Abbildung 5.1: Ansicht der Szenariokonfiguration¹⁰⁸

¹⁰⁸Eigene Abbildung

Mittels Rechtsklick im oberen Bereich kann das in Abbildung 5.2 gezeigte Kontextmenü aufgerufen werden. Darin finden sich Befehle zur Erstellung, zur Modifikation, zum Entfernen sowie zum Kopieren eines Szenarios.

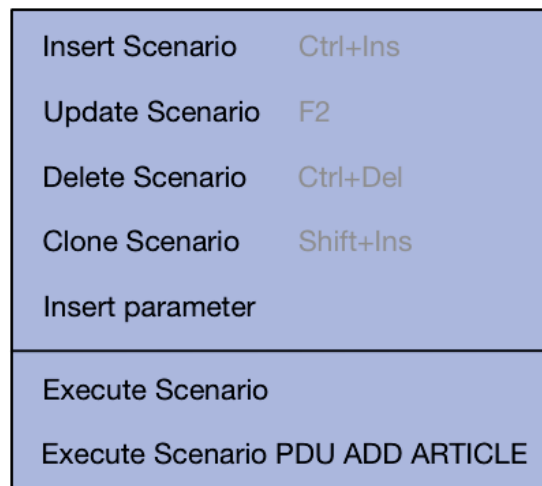


Abbildung 5.2: Kontextmenü in der Szenariokonfiguration¹⁰⁹

Die Anweisungen zum Verändern, Löschen bzw. zum Kopieren beziehen sich immer auf das aktuell ausgewählte Szenario. Für die Erstellung wird ein neuer Dialog aufgerufen, welcher in Abbildung 5.3 dargestellt ist. Wird ein Szenario als Pattern deklariert, so kann es im Nachhinein nicht mehr verändert werden.

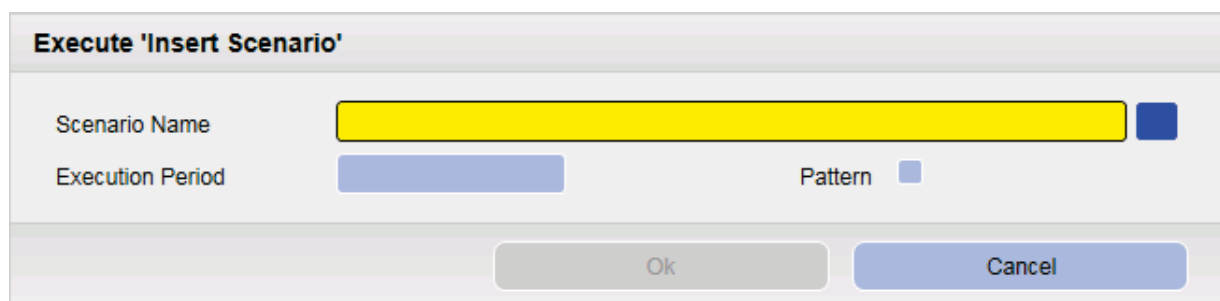


Abbildung 5.3: Dialog zur Erstellung eines neuen Szenarios¹¹⁰

¹⁰⁹Eigene Abbildung

¹¹⁰Eigene Abbildung

Weiters kann für ein ausgewähltes Szenario mittels „Insert Parameter“ ein neuer Parameter eingefügt werden, was den Aufruf des in Abbildung 5.4 gezeigten Eingabedialogs zur Folge hat. Die Auswahl der „Domain ID“ gibt an, in welchem Tab dieser Parameter angeordnet wird.

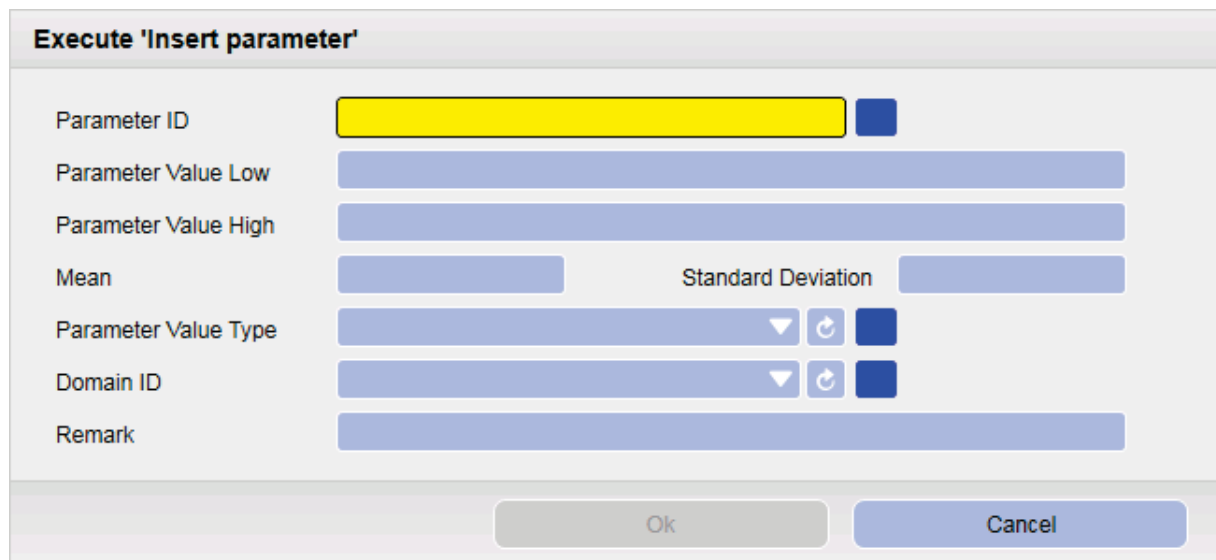


Abbildung 5.4: Dialog, um einen Parameter zu einem Szenario hinzuzufügen¹¹¹

Mittels „Execute Scenario“ kann schließlich das ausgewählte und zuvor konfigurierte Szenario ausgeführt werden. Dies ist auch über einen Button möglich, welcher in Abbildung 5.1 zu sehen ist. Zur Ankündigung neuer Artikeldaten, der in Abschnitt 4.3.2.12 beschriebenen Nachricht, muss kein eigenes Szenario erstellt werden, da hierfür keine Parameter notwendig sind. Deshalb gibt es zur Ausführung dieses Szenarios einen eigenen Befehl im Kontextmenü.

Auch im unteren Bereich gibt es ein Kontextmenü (siehe Abbildung 5.5), welches dazu dient, Parametereinträge abzuändern oder wieder zu entfernen.

Nachdem ein Szenario kreiert und anschließend ausgeführt wurde, werden von der Simulation die gewünschten Event-, Container-, FP- und WS-Einträge für die Nachrichten erstellt und in die entsprechenden Tabellen eingefügt. Wie die Ergebnisse angezeigt

¹¹¹Eigene Abbildung

werden und welche weiteren Möglichkeiten der Benutzer in dieser Ansicht hat, wird im folgenden Kapitel erläutert.

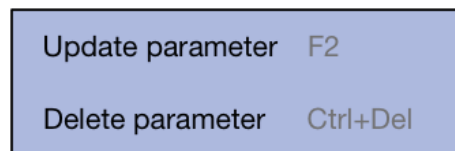


Abbildung 5.5: Kontextmenü für Parameter in der Szenariokonfiguration¹¹²

5.2.2 Darstellung der Simulationsdaten

Bereits in Kapitel 5.1.3 wurde erwähnt, dass eine eigene Anzeige zur gemeinsamen Darstellung der unterschiedlichen Nachrichtentabellen geschaffen wurde. Abbildung 5.6 zeigt diese Ansicht der Simulationsdaten.

Auf höchster Ebene werden die einzelnen Events der unterschiedlichen Szenarien dargestellt. Pro Szenariodurchlauf wird genau ein Event-Eintrag erstellt. Für das ausgewählte Event werden im Bereich darunter alle zugehörigen Container-Einträge angezeigt. Wieder eine Ebene unterhalb finden sich die FP-Einträge des markierten Containers und an letzter Stelle ist eine Liste aller SKUs für die im FP-Bereich ausgewählte Arbeitsstation angeführt.

Auch für diese Anzeige kann ein Filter angewandt werden, wobei hier nach verschiedensten Kriterien gefiltert werden kann, was Abbildung 5.7 zeigt. Je nach Einstellung werden nur die gewünschten Events und deren zugehörige Einträge angezeigt.

¹¹²Eigene Abbildung

Simulation data (Last update: 11:26:19)

Event ID	KNAPP HIF Action	KNAPP HIF Sub Act...	HOST Command	DAI Identifier	Import Status
87552	ADD	CONTAINER	NEW	ORD	5
87527	ADD	CONTAINER	DESLOT	STR	5
87537	ADD	CONTAINER	DESLOT	STR	5

Container

Event ID	WPO Number	Container Number	FP Count	Status	Error ID	Priority	WP Type ID
87552	0000407085	80709	1	10	0	51	CL
87552	0000407086	80710	1	10	0	29	CL

FP

Event ID	WPO Number	Container Number	Master FP Name	SKU Count	Status	Error ID	Priority
87552	0000407085	80709	OSR	2	10	0	0

SKU

Event ID	WPO Number	Container Number	Master FP Name	SKU Number	Status	Error ID	Priority
87552	0000407085	80709	OSR	1	10	0	0
87552	0000407085	80709	OSR	2	10	0	0

Abbildung 5.6: Ansicht der Simulationsdaten¹¹³

¹¹³Eigene Abbildung

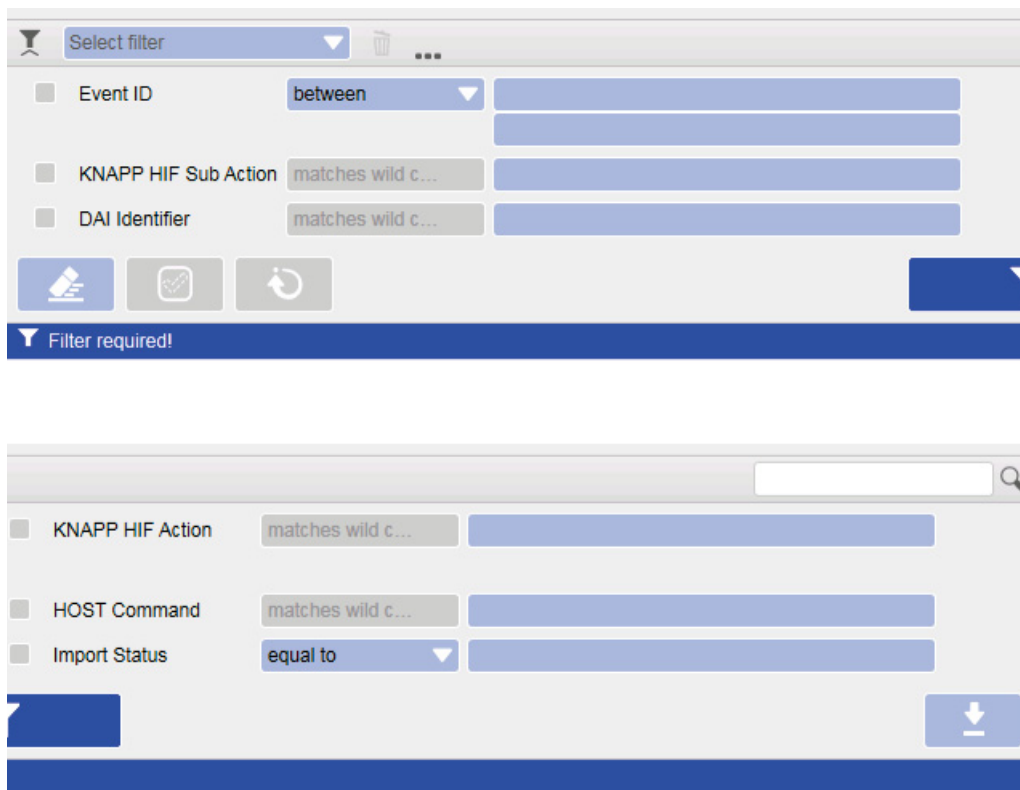


Abbildung 5.7: Filter zur Einschränkung der Anzeige der Simulationsdaten¹¹⁴

Durch Aufruf des Kontextmenüs (siehe Abbildung 5.8) im Event-Bereich können Events mit allen zugehörigen Einträgen wieder gelöscht werden. Außerdem ist es möglich, einzelne Nachrichten von hier aus ans WCS zu transferieren.

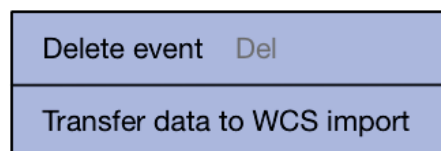


Abbildung 5.8: Kontextmenü für Events¹¹⁵

¹¹⁴Eigene Abbildung

¹¹⁵Eigene Abbildung

Auch im Container-Bereich ist ein Kontextmenü vorhanden, in dem sich sechs weitere Befehle befinden. Dieses ist in Abbildung 5.9 dargestellt. Von hier aus können die Prozeduren zur Erstellung jener Nachrichten aufgerufen werden, welche zum Löschen oder Verändern einzelner Aufträge verwendet werden. Befehle, die für den selektierten Auftrag nicht ausgeführt werden können, werden ausgegraut angezeigt und sind nicht auswählbar. Da es sich im dargestellten Fall um einen Warenausgang handelt, kann auch nur dieser storniert werden. Mit „Mod Container“ wird die Nachricht zur Veränderung eines Kommissionierauftrags (siehe Abschnitt 4.3.2.6) erstellt. Durch Auswahl dieses Befehls öffnet sich der in Abbildung 5.10 dargestellte Eingabedialog, in dem die Cut-off-Time, das Lieferunternehmen sowie die Priorität aktualisiert werden können. Mittels „Forward To WCS“ kann die Nachricht nach ihrer Erstellung sofort ans WCS übermittelt werden.

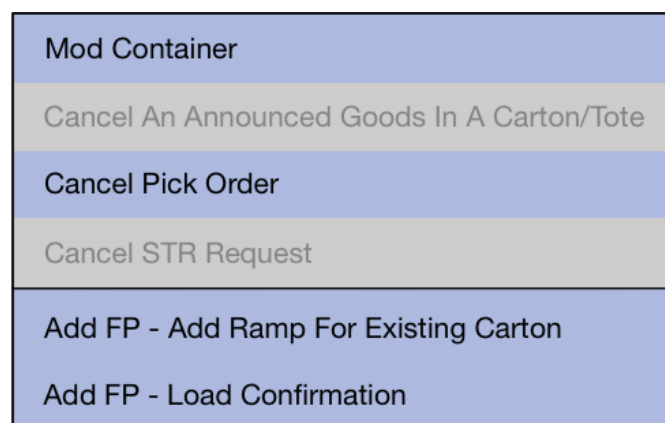


Abbildung 5.9: Kontextmenü für Container¹¹⁶

Im unteren Bereich des Kontextmenüs finden sich außerdem die Aufrufe zum Hinzufügen bzw. zur Bestätigung einer Rampe, die in den Abschnitten 4.3.2.10 und 4.3.2.11 genau erläutert wurden. Der Name der Rampe muss im zugehörigen Eingabedialog aus einer Liste zur Verfügung stehender Rampen ausgewählt werden. Außerdem können auch diese Nachrichten nach ihrer Erstellung sofort ans WCS transferiert werden. Abbildung 5.11 zeigt den Eingabedialog zum Hinzufügen einer Rampe, jener zur Bestätigung einer bereits bestehenden Rampe sieht allerdings nahezu ident aus.

¹¹⁶Eigene Abbildung

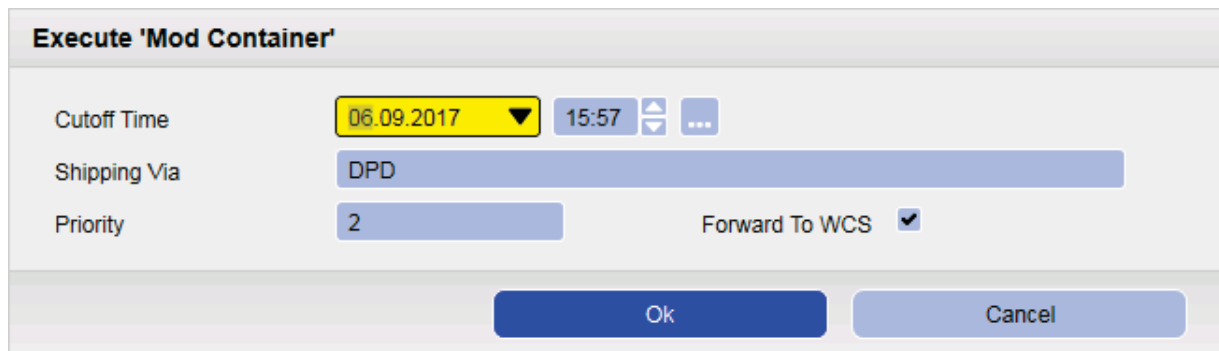


Abbildung 5.10: Eingabedialog zur Aktualisierung eines Kommissionierauftrags¹¹⁷

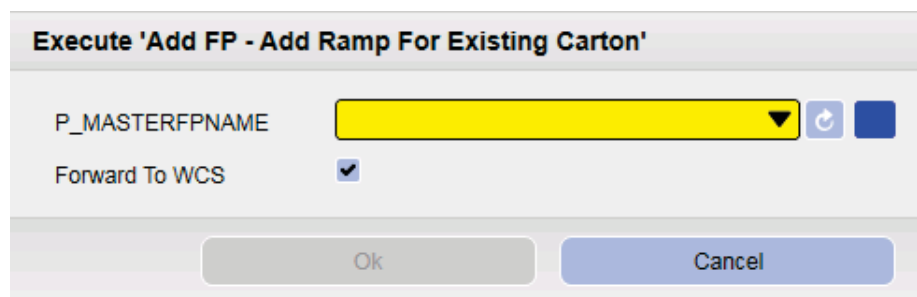


Abbildung 5.11: Eingabedialog zum Hinzufügen einer Rampe¹¹⁸

Um die durch diese Befehle erstellten Nachrichten zu betrachten, kann allerdings nicht die Standardansicht der Simulationsdaten verwendet werden. Denn diese Nachrichten verfügen über keinen Container-Eintrag, weshalb eine eigene Ansicht geschaffen wurde, in welcher der FP-Eintrag direkt mit dem Event-Eintrag verknüpft ist. Diese Ansicht ist in Abbildung 5.12 dargestellt.

Im unteren Bereich dieser Anzeige sind zwei Tabs zu sehen, betitelt mit „FP“ und „Stock“. Ersterer ist für FP-Einträge zuständig, wie eben beschrieben wurde, zweiterer stellt Einträge des Lagerbestands dar. Diese entstehen bei der Ausführung der Nachricht zur Sperrung bzw. Freigabe einzelner Artikel (siehe Abschnitt 4.3.2.13). Wie solche Nachrichten mittels dieser Simulation erstellt werden, wird durch die Erläuterungen des folgenden Kapitels aufgezeigt.

¹¹⁷Eigene Abbildung

¹¹⁸Eigene Abbildung

Simulation Data FP Stock x +
Simulation Data FP Stock
 Last update: 12:01:07
 Row 4 of 7

Event ID	KNAPP HIF Action	KNAPP HIF Sub Act...	HOST Command	DAI Identifier	Import Status
87552	ADD	CONTAINER	NEW	ORD	5
87527	ADD	CONTAINER	DESLOT	STR	5
87537	ADD	CONTAINER	DESLOT	STR	5
87528	ADD	FP	CONSOL	TMR	5
87532	ADD	FP	CONSOL	TMR	5
87533	ADD	CONTAINER	DESLOT	STR	5
87551	ADD	CONTAINER	ADD	IGD	5

Drag a column header here to group by that column

FP Stock

Event ID	WPO Number	Container Number	Master FP Name	SKU Count	Status	Error ID	Priority
87528	0000406978	80638	DP04	0	10	0	

Row 1 of 1

Abbildung 5.12: Ansicht der Simulationsdaten für Nachrichten ohne Container-Eintrag¹¹⁹

¹¹⁹Eigene Abbildung

5.2.3 Anzeige der Artikel Tabellen

In den beiden vorherigen Kapitel wurde zuerst die Szenariokonfiguration erläutert und anschließend die Darstellung der Simulationsdaten aufgezeigt. Eine weitere Ansicht ist zur umfassenden Bedienung der Simulation notwendig und diese zeigt alle zur Verfügung stehenden Artikel (siehe Abbildung 5.14). Wichtig sind in dieser Tabelle vor allem die Artikelnummer sowie die Artikelbeschreibung. Auf diese beiden Spalten kann, wie auch in den bereits beschriebenen Anzeigen, ein Filter angewandt werden.

Diese Ansicht wird außerdem dazu verwendet, um einzelne Artikel zu sperren oder wieder freizugeben. Über das in Abbildung 5.13 dargestellte Kontextmenü kann die entsprechende Nachricht mittels „Mod Stock“ erstellt werden. Weiters können alle Artikeldaten vom WCS importiert werden, um die Tabelle zu aktualisieren.

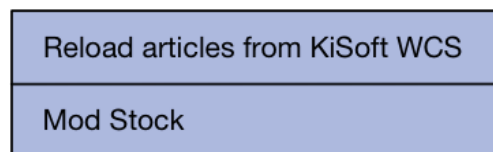


Abbildung 5.13: Kontextmenü für Artikel¹²⁰

Nachdem in diesem Kapitel die allgemeine Bedienung der Simulation erläutert wurde, werden im nächsten Kapitel alle notwendigen Schritte zur Erzeugung der Testdaten an einem expliziten Beispiel durchgespielt.

¹²⁰Eigene Abbildung

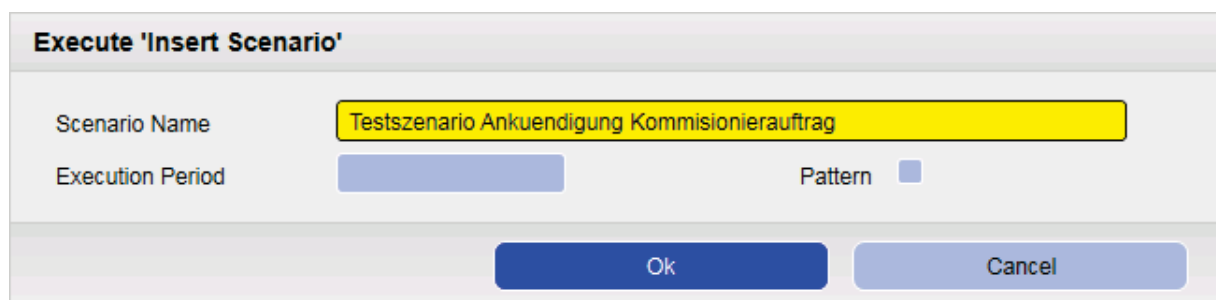
Item Number	Area	Item Description	Full Case Quantity
V59394550	G2P	O-CL NYLON SLIM POP BLUE/WHT/SLVR/PINK	12
V59394560	G2P	O-CL NYLON SLIM POP BLUE/WHT/SLVR/PINK	12
V59394570	G2P	O-CL NYLON SLIM POP BLUE/WHT/SLVR/PINK	12
Z82740480	G2P	O-DT GRPH CREW T WHT	50
Z82740500	G2P	O-DT GRPH CREW T WHT	50
Z82740520	G2P	O-DT GRPH CREW T WHT	50
Z82740540	G2P	O-DT GRPH CREW T WHT	50
Z82742330	G2P	O-CF MATT SHORT ULTYE	50
Z82742350	G2P	O-CF MATT SHORT ULTYE	50
Z82749310	G2P	O-CF PERF BOSHRT2 CONBLU	50
Z82749330	G2P	O-CF PERF BOSHRT2 CONBLU	50
Z82749350	G2P	O-CF PERF BOSHRT2 CONBLU	50
Z82749370	G2P	O-CF PERF BOSHRT2 CONBLU	50
Z82750310	G2P	O-CF PERF SHRT2A RBNACA	50
Z82750330	G2P	O-CF PERF SHRT2A RBNACA	50
Z82750350	G2P	O-CF PERF SHRT2A RBNACA	50
Z82750370	G2P	O-CF PERF SHRT2A RBNACA	50
Z82751310	G2P	O-CF PERF BOSHRT3 CONBLU	50
Z82751330	G2P	O-CF PERF BOSHRT3 CONBLU	50
Z82751350	G2P	O-CF PERF BOSHRT3 CONBLU	50
Z82751370	G2P	O-CF PERF BOSHRT3 CONBLU	50
Z82752310	G2P	O-CF PERF BOSHRT3 EXCRED	50
Z82752330	G2P	O-CF PERF BOSHRT3 EXCRED	50
Z82752350	G2P	O-CF PERF BOSHRT3 EXCRED	50
Z82752370	G2P	O-CF PERF BOSHRT3 EXCRED	50
Z82755310	G2P	O-CF PERF BOSH03B PNKFUS	50
Z82755330	G2P	O-CF PERF BOSH03B PNKFUS	50
Z82755370	G2P	O-CF PERF BOSH03B PNKFUS	50

Abbildung 5.14: Artikeltabelle¹²¹

¹²¹Eigene Abbildung

5.3 Simulationsdurchlauf anhand eines konkreten Beispiels

Das Ziel dieses Simulationsdurchlaufs ist die Erstellung einer Nachricht zur Ankündigung eines Kommissionierauftrags, wie sie in Abschnitt 4.3.2.4 beschrieben wurde. Dem WCS soll also mitgeteilt werden, dass ein Auftrag mit bestimmten Artikeln abgearbeitet werden soll. Dafür muss in der Szenariokonfiguration mittels „Insert Scenario“ ein entsprechendes Szenario erstellt werden, was in Abbildung 5.15 gezeigt wird. Hierfür ist ausschließlich der Name des Szenarios einzutragen. Das Feld „Execution Period“ wird nur für Szenarien benötigt, die automatisch in regelmäßigen Abständen ausgeführt werden sollen. Da dieses Szenario außerdem nicht als Vorlage für andere dienen soll, wird „Pattern“ nicht ausgewählt.



The image shows a dialog box titled "Execute 'Insert Scenario'". It contains two input fields: "Scenario Name" and "Execution Period". The "Scenario Name" field is highlighted in yellow and contains the text "TestszENARIO Ankuendigung Kommisionierauftrag". The "Execution Period" field is empty. To the right of the "Execution Period" field is a "Pattern" checkbox, which is currently unchecked. At the bottom of the dialog box, there are two buttons: "Ok" and "Cancel".

Abbildung 5.15: Erstellung eines Szenarios¹²²

Anschließend kann damit begonnen werden, alle notwendigen Parameter für das erstellte Szenario zu generieren. Zuerst wird der Typ des Szenarios definiert, d.h. es wird festgelegt, dass es sich um die Ankündigung eines Kommissionierauftrags handelt. Dies ist in Abbildung 5.16 dargestellt. Die Zeichenkette „ORD (ADD CONTAINER)“ muss genau auf diese Weise eingetragen werden, um eine Identifizierung der Nachrichtenart zu ermöglichen. Dieser Parameter betrifft das gesamte Szenario, weshalb er im

¹²²Eigene Abbildung

Szenario-Tab angeordnet werden soll und dies als „Domain ID“ ausgewählt wird. Im Feld „Parameter Value Type“ wird vermerkt, dass nur ein einzelner Wert übergeben wird.

Field	Value
Parameter ID	SCENARIO_TYPE
Parameter Value Low	ORD (ADD CONTAINER)
Parameter Value High	
Mean	
Standard Deviation	
Parameter Value Type	S
Domain ID	SCENARIO
Remark	

Abbildung 5.16: Erstellung eines Parameters für die Ankündigung eines Kommissionierauftrags¹²³

Auf die gleiche Art und Weise werden alle übrigen Parameter erstellt. Jene der Szenario-Kategorie sind in Abbildung 5.17 aufgelistet. Mit ORDER_COUNT wird die Anzahl der Aufträge bzw. Container angegeben. In diesem Beispiel wird eine Zahlenspanne übergeben, aus der ein zufälliger Wert ausgewählt werden soll. Das erstellte Event soll demnach ein oder zwei Container-Einträge beinhalten.

Das Feld SINGLE_PCS_LNS übermittelt eine Liste potentieller Optionen und SINGLE_PCS_LNS_DIST führt die Wahrscheinlichkeit für jede dieser Möglichkeiten an. Genauer gesagt müssen Prozentgrößen für die Wahrscheinlichkeiten in gleicher Reihenfolge wie die entsprechenden Werte angegeben werden und dies in kumulativer Form. Für dieses Beispiel werden die ersten beiden Einträge demnach mit einer Wahrscheinlichkeit gleich null ausgewählt und der letzte Wert zu 100 Prozent. Alle Aufträge

¹²³Eigene Abbildung

verwenden somit den Wert „R“. Dies muss dem System auf diese Weise mitgeteilt werden, da dieses Feld so definiert ist, dass lediglich eine Liste übergeben werden kann und nicht nur eine einzelne Zeichenkette.

Scenario ID	Parameter ID	Parameter Value Low	Parameter Value High	Parameter Value Type
648	ORDER_COUNT	1	2	R
648	SCENARIO_TYPE	ORD (ADD CONTAINER)		S
648	SINGLE_PCS_LNS	SP;SL;R		D
648	SINGLE_PCS_LNS_DIST	0;0;100		L

Abbildung 5.17: Szenario-Parameter für die Ankündigung eines Kommissionierauftrags¹²⁴

Auch auf die Bedeutung dieser Werte für das gewählte Beispiel soll kurz eingegangen werden. Dieses Feld definiert, wieviel Prozent der Aufträge aus mehreren Artikeln bestehen und wie viele nur aus einem. „SP“ steht für „Single Piece“ und bezeichnet jene Aufträge, welche nur einen Artikeltyp und davon nur ein Stück beinhalten. „Single Lines“ werden durch „SL“ abgekürzt und bestehen ebenso nur aus einer Art von Artikel. Die Anzahl kann allerdings höher sein und wird durch einen Parameter der SKU-Kategorie definiert, der anschließend noch genauer erklärt wird. Alle übrigen Aufträge sind vom Typ „R“. Die entsprechenden Größen werden ebenfalls durch Parameter der SKU-Kategorie vorgegeben.

Eine Tabelle mit allen notwendigen Parametern der Container-Kategorie wird in Abbildung 5.18 dargestellt. Einige davon sollen nun näher beschrieben werden, um die Vorgehensweise bei der Erstellung dieser Parameter deutlich zu machen.

Als weiteres Beispiel für die Übermittlung einer Liste mit zugehörigen Wahrscheinlichkeiten, soll das Destinationsland verwendet werden. Die hierfür verwendeten Parameter sind DEST_CNTRY und DEST_CNTRY_DIST. In der erstellten Nachricht soll ein Container mit einer Wahrscheinlichkeit von 33% aus Deutschland stammen, zu 47%

¹²⁴Eigene Abbildung

Generierung von Testdaten für Materialflusssteuerungen im Bereich der Intralogistik durch den Einsatz einer Host-Simulation

Kapitel 5. Durchführung der Tests mittels Simulation

aus Österreich und zu 20% aus Großbritannien. Auch in einigen weiteren Feldern werden Listen übergeben, in CUTOFF_UPD_AL z.B. eine Liste, die nur aus „Ja“ oder „Nein“ besteht, wobei „Ja“ zu 70% ausgewählt wird.

Im Feld PRIORITY wird wiederum eine Zahlenspanne übermittelt, so dass ein zufälliger Wert von 1 bis 99 verwendet wird.

Scenario ID	Parameter ID	Parameter Value Low	Parameter Value High	Parameter Value Type
648	CONTTYPEID	E01;E02;E03;E04;EXS		D
648	CONTTYPEID_DIST	20;40;60;80;100		L
648	CUTOFF_TIME	1000	1000	R
648	CUTOFF_UPD_AL	Y;N		D
648	CUTOFF_UPD_AL_DIST	70;100		L
648	DEST_CNTRY	DE;AT;UK		D
648	DEST_CNTRY_DIST	33;80;100		L
648	EXPORT_ORDER	Y;N		D
648	EXPORT_ORDER_DIST	50;100		L
648	PRIORITY	1	99	R
648	SHIP_VIA	DHL;DPD		D
648	SHIP_VIA_DIST	80;100		L
648	VAS_CODE_CONT	10M;20M		D
648	VAS_CODE_CONT_DIST	50;100		L

Abbildung 5.18: Container-Parameter für die Ankündigung eines Kommissionierauftrags¹²⁵

Für die FP-Kategorie sind für diese Art von Nachricht keine Parameter zu definieren. Alle benötigten SKU-Parameter werden in Abbildung 5.19 angeführt.

Scenario ID	Parameter ID	Parameter Value Low	Parameter Value High	Parameter Value Type
648	ITEMDESC	%		S
648	ITEMNO	S%		I
648	QTY_ITEM	2	4	R
648	QTY_PIECES_ITEM	2	3	R
648	STOCKCHECK	True		B
648	VAS_CODE_WS	MUV		S

Abbildung 5.19: SKU-Parameter für die Ankündigung eines Kommissionierauftrags¹²⁶

¹²⁵Eigene Abbildung

¹²⁶Eigene Abbildung

Zur Einschränkung der Artikelauswahl können in ITEMDESC Kriterien für die Artikelbeschreibung und in ITEMNO für die Artikelnummer angegeben werden. In diesem Beispiel kann die Artikelbeschreibung beliebig sein, was durch das Prozentsymbol ausgedrückt wird. Die Artikelnummer hingegen muss mit einem „S“ beginnen, darauf können allerdings beliebige Zeichen folgen.

In QTY_PIECES_ITEM wird die gewünschte Stückzahl je Artikel angegeben, falls der Auftrag nicht vom Typ „SP“ ist, da jene Aufträge, wie bereits erwähnt, unabhängig von diesem Parameter immer nur aus einem Stück eines Artikels bestehen. Für Aufträge des Typs „R“ wird außerdem der Parameter QTY_ITEM benötigt, welcher die Anzahl unterschiedlicher Artikel angibt. Da das erstellte Szenario nur Aufträge dieses Typs enthält, sollen jeweils zwei bis vier verschiedene Artikel und von jedem Artikel entweder zwei oder drei Stück ausgewählt werden. Allerdings wird die Obergrenze für die Stückzahl durch den Lagerbestand gesetzt, falls für den Parameter STOCKCHECK „True“ übergeben wird. In diesem Fall kann die tatsächliche Anzahl geringer sein als durch QTY_PIECES_ITEM vorgegeben.

Nachdem alle nötigen Parameter definiert wurden und das Szenario demnach fertig erstellt ist, kann es über den Button „Execute Scenario“ in der Szenariokonfiguration (siehe Abbildung 5.1) ausgeführt werden. In der Anzeige der Simulationsdaten können nun alle Einträge der eben erstellten Nachricht eingesehen werden. Da alle Events mit einer fortlaufenden ID versehen werden, sind die gerade erzeugten Einträge mit der derzeit höchsten ID gekennzeichnet. Abbildung 5.20 zeigt die Ansicht der Simulationsdaten für das durchgeführte Beispielszenario.

Hier kann überprüft werden, ob die Einträge den vorgegeben Parametern zufolge korrekt erstellt wurden. Das Event beinhaltet, wie gewünscht, zwei Container-Einträge. Als Destinationsland ist bei einem der Aufträge „AT“ und beim anderen „DE“ eingetragen, was ebenfalls den Vorgaben entspricht. Ob die genauen Wahrscheinlichkeiten erfüllt wurden, kann natürlich nur bei einer weit größeren Menge an Aufträgen überprüft werden. Auch die Priority jedes Auftrags befindet sich innerhalb der gewünschten Spanne von 1 bis 99.

Simulation data (Last update: 18:07:28)

Event ID	KNAPP HIF Action	KNAPP HIF Sub Act...	HOST Command	DAI Identifier	Import Status
87628	ADD	CONTAINER	NEW	ORD	5

Container

Event ID	WPO Number	Container Number	FP Count	Shipping Via	Destination Country	Priority	Status
87628	0000407212	80834	1	DHL	AT	30	10
87628	0000407213	80835	1	DHL	DE	19	10

FP

Event ID	WPO Number	Container Number	Master FP Name	SKU Count	Status	Error ID	Priority
87628	0000407212	80834	OSR	4	10	0	0

SKU

Event ID	WPO Number	Container Number	Master FP Name	SKU Number	Item Number	Quantity	Status	Error
87628	0000407212	80834	OSR	1	S97063540	3	10	0
87628	0000407212	80834	OSR	2	S82931630	3	10	0
87628	0000407212	80834	OSR	3	S83382600	2	10	0
87628	0000407212	80834	OSR	4	S79665600	3	10	0

Abbildung 5.20: Simulationsdaten für die Ankündigung eines Kommissionierauftrags¹²⁷

¹²⁷Eigene Abbildung

Im FP-Eintrag findet sich der Name der Zielstation dieser Nachricht. Wie erwartet ist dies für einen Warenausgang das „OSR“.

Der unterste Bereich zeigt die unterschiedlichen Artikel. Insgesamt sind hier vier Einträge zu sehen, was innerhalb der vorgegebenen Spanne liegt. Jede der Artikelnummern beginnt mit einem „S“ und die jeweilige Stückzahl liegt stets bei zwei oder drei.

Für dieses Beispiel konnte also gezeigt werden, dass die Nachrichten entsprechend der vom Benutzer generierten Vorgaben erstellt wurden. Das hierbei kreierte Szenario muss keinesfalls verworfen werden, sondern kann beliebig oft ausgeführt werden, um weitere Nachrichten zu erstellen. Sind geringe Änderungen notwendig, so müssen hierfür nur die entsprechenden Parameter modifiziert werden. Das Szenario kann davor kopiert werden, um auch die ursprünglichen Einstellungen beizubehalten und später erneut verwenden zu können.

6 Conclusio und Ausblick

Wie in Kapitel 5 gezeigt wurde, können mittels der Host-Simulation in geringer Zeit die benötigten Daten zur Durchführung der Tests generiert werden. Ein einmal erstelltes Szenario kann beliebig oft ausgeführt werden und somit ist es möglich, per Knopfdruck eine Nachricht mit der gewünschten Anzahl an Aufträgen zu generieren. Bei manueller Erzeugung der Testdaten muss insgesamt weit mehr Zeit aufgewandt werden, weil jede Nachricht einzeln erstellt werden muss. Mit Hilfe der Simulation reichen hingegen wenige Schritte aus, um eine große Menge an Nachrichten zu generieren. Dadurch ist ein immenser Zeitgewinn im Vergleich zu anderen Varianten zur Erstellung der Testdaten möglich.

Der anfängliche Zeitaufwand ist höher als bei manueller Erstellung der Testdaten, da ein Benutzer zuerst angelernt werden muss, damit er die Simulation richtig bedienen kann. Doch nach Beendigung der Einschulung ist der Zeitgewinn groß, da beliebig viele Testnachrichten in kurzer Zeit erstellt werden können. Der Testprozess wurde durch die Host-Simulation somit erheblich beschleunigt sowie in Bezug auf Massentests überhaupt erst ermöglicht.

Die im Rahmen dieser Arbeit durchgeführte Simulation wurde auf Basis eines Projekts erstellt und erhebt in der jetzigen Form keinen Anspruch auf Allgemeingültigkeit. Die spezifische Natur der generierten Software führt dazu, dass sie ohne vorangehende Adaptionen für andere ähnliche Projekte nicht verwendet werden kann. Um eine Anpassung der Simulation für ein neues Projekt zu ermöglichen, ist demnach die Arbeit eines Softwareentwicklers und ein hoher Zeitaufwand nötig, weshalb diese Vorgehensweise keinen generell günstigen Weg darstellt.

Diese Arbeit soll dazu verwendet werden, das Fundament für die Entwicklung eines generischen Konzepts zur Erstellung solcher Host-Simulationen zu schaffen. Das Ziel ist, dass genaue Strukturen für unterschiedliche Projekte, wie z.B. der Aufbau einer Nachricht, nicht mehr durch einen Softwareentwickler implementiert werden müssen, sondern von einem geschulten Benutzer über eine geeignete Oberfläche definiert werden können. Um beispielsweise den Nachrichtenaufbau zu konfigurieren, soll der Anwender festlegen können, welche Einträge vorhanden sind, wie diese benannt werden und wie sie miteinander verknüpft sind. Für die Szenariokonfiguration sollen die mögliche Aufteilung der Parameter sowie die nötigen Felder zur Erstellung von Szenarien und ihren zugehörigen Parametern definiert werden können.

Dieser Bereich bietet noch eindeutigen Forschungsbedarf. Vor allem die Unterscheidung zwischen einzelnen Typen von Nachrichten ist ein Problem, das auf andere Weise gelöst werden müsste als bisher. Ein Ansatz hierfür wäre beispielsweise die Erstellung weiterer Tabellen in der Datenbank, in welchen die Definitionen des Benutzers abgelegt werden. Eine Prozedur müsste anschließend diese Informationen extrahieren und zur Differenzierung zwischen unterschiedlichen Nachrichten verwenden.

Die obigen Ausführungen dienen dazu, einen Einblick in potentielle Fortsetzungen zu geben, welche auf Basis der im Zuge dieser Arbeit erstellten Host-Simulation ermöglicht wurden.

Literaturverzeichnis

Buch- und Journalquellen

Bichler, K, Krohn, R und Philippi, P (2005). *Gabler Kompakt-Lexikon Logistik: 1.800 Begriffe nachschlagen, verstehen, anwenden*. 1. Auflage. SpringerLink : Bücher. Gabler Verlag. ISBN: 9783322964854.

<https://books.google.at/books?id=XsEdBAAAQBAJ>

Bungartz, H J et al. (2009). *Modellbildung und Simulation*. EXamen. press Series. Springer Berlin Heidelberg. ISBN: 9783540798101.

<https://books.google.at/books?id=hNOFEnE4z7oC>

Daniluk, D. und Chisu, R. (2010). „Simulation und Emulation im Internet der Dinge“. In: *Internet Der Dinge in Der Intralogistik*. Hrsg. von W. A. Günthner und M. Ten Hompel. VDI-Buch. Springer Berlin Heidelberg, S. 147–164. ISBN: 9783642048968.

<https://books.google.at/books?id=zr1CQcyEymYC>

Eley, M (2012). *Simulation in der Logistik: Einführung in die Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges Plant Simulation*. Springer-Lehrbuch. Springer Berlin Heidelberg. ISBN: 9783642273728.

https://books.google.at/books?id=tG94ghAoa%7B%5C_%7DoC

Feuerstein, S. und Pribyl, B. (2009). *Oracle PL/SQL Programming*. 5th Edition. Animal Guide. O'Reilly Media. ISBN: 9781449379384.

<https://books.google.at/books?id=iWhbFeboD84C>

Fujimoto, R M (2000). *Parallel and distributed simulation systems*. Wiley series on parallel and distributed computing. Wiley. ISBN: 9780471183839.

https://books.google.at/books?id=b%7B%5C_%7DdQAAAAMAAJ

Hall, A und Fagen, R (1956). *Definition of system*. Techn. Ber.

Haußner, C., Elger, J. und Kuzmany, F. (2010). „Der Lebenszyklus heutiger Materialflusssysteme - eine Übersicht“. In: *Internet Der Dinge in Der Intralogistik*. Hrsg. von W A Günthner und M Ten Hompel. VDI-Buch. Springer Berlin Heidelberg, S. 167–180. ISBN: 9783642048968.

<https://books.google.at/books?id=zr1CQcyEymYC>

Hrdliczka, V. et al. (1997). *Leitfaden für Simulationsbenutzer in Produktion und Logistik*. ASIM-Mitteilungen aus den Arbeitskreisen. ASIM.

<https://books.google.at/books?id=6AjUPgAACAAJ>

Jünemann, R. und Beyer, A. (1998). *Steuerung von Materialfluss- und Logistiksystemen: Informations- und Steuerungssysteme, Automatisierungstechnik*. Logistik in Industrie, Handel und Dienstleistungen. Springer Berlin Heidelberg. ISBN: 9783642722257.

<https://books.google.at/books?id=8gsgBgAAQBAJ>

Klaus, P und Krieger, W (2008). *Gabler Lexikon Logistik*. 4. Auflage. Gabler Verlag.
ISBN: 9783834987723.

<https://books.google.at/books?id=5XS7Khx9kfgC>

Krämer, K. (2002). *Automatisierung in Materialfluss und Logistik: Ebenen, Informationslogistik, Identifikationssysteme, intelligente Geräte*. Deutscher Universitätsverlag.
ISBN: 9783322812216.

<https://books.google.at/books?id=p4r0BQAAQBAJ>

Kuhn, A und Rabe, M (1998). *Simulation in Produktion und Logistik: Fallbeispielsammlung*. Springer Berlin Heidelberg. ISBN: 9783642720680.

<https://books.google.at/books?id=1DeGBwAAQBAJ>

Nieke, C. (2010). „Materialflussteuerung heute und ihre Defizite“. In: *Internet Der Dinge in Der Intralogistik*. Hrsg. von W. A. Günthner und M. Ten Hompel. VDI-Buch. Springer Berlin Heidelberg, S. 15–21. ISBN: 9783642048968.

<https://books.google.at/books?id=zr1CQcyEymYC>

Rabe, M, Spieckermann, S und Wenzel, S (2008). *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*. VDI-Buch. Springer Berlin Heidelberg. ISBN: 9783540352815.

https://books.google.at/books?id=3JFV%7B%5C_%7DV5FZ1YC

Ten Hompel, M. und Heidenblut, V. (2006). *Taschenlexikon Logistik: Abkürzungen, Definitionen und Erläuterungen der wichtigsten Begriffe aus Materialfluss und Logistik*. VDI-Buch. Springer Berlin Heidelberg. ISBN: 9783540285816.

<https://books.google.at/books?id=PjysAAAACAAJ>

Ten Hompel, M. und Schmidt, T. (2006). *Warehouse Management - Organisation und Steuerung von Lager- und Kommissioniersystemen*. 2. Auflage. VDI-Buch. Springer Berlin Heidelberg. ISBN: 9783540270218.

<https://books.google.at/books?id=PbvOgfuY0w0C>

Ten Hompel, M. und Schmidt, T. (2010). *Warehouse Management - Organisation und Steuerung von Lager- und Kommissioniersystemen*. 4. Auflage. VDI-Buch. Springer Berlin Heidelberg. ISBN: 9783642031854.

<https://books.google.at/books?id=NedyoESLU40C>

Verriet, J. und Van Wijngaarden, B. (2011). „A Reference Architecture Capturing Structure and Behaviour of Warehouse Control“. In: *Automation in Warehouse Development*. Hrsg. von R. Hamberg und J. Verriet. SpringerLink : Bücher. Springer London, S. 17–32. ISBN: 9780857299680.

<https://books.google.at/books?id=DYCAQ64h0soC>

Wenzel, S. und Peter, T. (2017). *Simulation in Produktion und Logistik 2017*. Kassel University Press. ISBN: 9783737601924.

<https://books.google.at/books?id=eT83DwAAQBAJ>

Wenzel, S., Weiß, M. et al. (2007). *Qualitätskriterien für die Simulation in Produktion und Logistik: Planung und Durchführung von Simulationsstudien*. VDI-Buch. Springer Berlin Heidelberg. ISBN: 9783540352761.

<https://books.google.at/books?id=3ykiBAAAQBAJ>

Interne Dokumente

Specification - Functional (2017).

[http://polarion:8080/polarion/#/project/DE484/wiki/Specification/
PH_DE484-FU_en](http://polarion:8080/polarion/#/project/DE484/wiki/Specification/PH_DE484-FU_en)

(Aufruf am 15.10.2017)

Specification - Interface (2017).

[http://polarion:8080/polarion/#/project/DE484/wiki/Specification/
PH_DE484-IN_en](http://polarion:8080/polarion/#/project/DE484/wiki/Specification/PH_DE484-IN_en)

(Aufruf am 15.10.2017)

Anhang

Der Anhang beinhaltet den im Zuge dieser Arbeit erstellten Quellcode zur Generierung der unterschiedlichen Nachrichten:

```
1  /*
2  <type>          proc
3  <name>          GenerateOrders
4  <short>         Generates orders
5  <author>       Herka Jakob
6  */
7  PROCEDURE GenerateOrders (pnu_RetCode   OUT NUMBER,
8                          p_ScenarioId   IN ACT_SCENARIO.Scenario_Id%TYPE,
9                          p_ImpStatus    IN SIMULHOST_Event.Imp_Status%TYPE := GENERIC_IMP_STAT_GENERATED)
10 IS
11   vvc_Module          LOG_ENTRIES.SENDER%TYPE := gvc_ModuleHeader || 'GenerateOrders';
12   vnu_ContainerCount  INTEGER;
13   vnu_ContainerIter   INTEGER;
14   vnu_WsCount         INTEGER;
15   vnu_WsIter          INTEGER;
16   vnu_WsQtyIter       INTEGER;
17   v_ScenarioType      ACT_SCENARIO_PAR.PAR_VALUE_LOW%TYPE;
18   v_SinglePcsLines    ACT_SCENARIO_PAR.PAR_VALUE_LOW%TYPE;
19   v_Stockcheck        ACT_SCENARIO_PAR.PAR_VALUE_LOW%TYPE;
20
21   setWsQty            BOOLEAN;
22   isStockcheck        BOOLEAN;
23   v_RowCount          INTEGER;
24   v_Offset            INTEGER;
25   v_CurrentStock      NUMBER;
26   v_VarChar           VARCHAR2(255);
27   v_StrSeperator      VARCHAR2(10);
28   v_QtyReturnStockY   INTEGER;
29   v_QtyReturnStockN   INTEGER;
30
31   --Event
32   v_EventId           SIMULHOST_Event.ID%TYPE;
33   v_DaiIdentifier     SIMULHOST_Event.DAI_IDENTIFIER%TYPE;
34   v_Action            SIMULHOST_Event.ACTION%TYPE;
35   v_SubAction         SIMULHOST_Event.SUB_ACTION%TYPE;
36   v_Command           SIMULHOST_Event.COMMAND%TYPE;
37
38   --Container
39   v_WpoNo             SIMULHOST_CONTAINER.WpoNo%TYPE;
40   v_CntNo             SIMULHOST_CONTAINER.CntNo%TYPE;
41   v_FpCount           SIMULHOST_CONTAINER.FP_CNT%TYPE := 1;
42   v_StatusCont        SIMULHOST_CONTAINER.STATUS%TYPE;
43   v_ErrorId           SIMULHOST_CONTAINER.ERRORID%TYPE := 0;
```

```
44 v_PriorityCont      SIMULHOST_CONTAINER. Priority%TYPE;
45 v_WpTypeld         SIMULHOST_CONTAINER. WpTypeld%TYPE;
46 v_Licenceplate     SIMULHOST_CONTAINER. Licenceplate%TYPE;
47 v_ContTypeld       SIMULHOST_CONTAINER. ContTypeld%TYPE;
48 v_VasCodeCont      SIMULHOST_CONTAINER. VAS_CODE%TYPE;
49 v_CutoffTime       SIMULHOST_CONTAINER. CUTOFF_TIME%TYPE;
50 v_ReturnReference   SIMULHOST_CONTAINER. RETURN_REFERENCE%TYPE;
51 v_ShipVia           SIMULHOST_CONTAINER. SHIP_VIA%TYPE;
52 v_ExportOrder      SIMULHOST_CONTAINER. EXPORT_ORDER%TYPE;
53 v_CustomerOrderTime SIMULHOST_CONTAINER. CUSTOMER_ORDER_TIME%TYPE;
54 v_WmsStartTime     SIMULHOST_CONTAINER. WMS_START_TIME%TYPE;
55 v_RepackContTypeld SIMULHOST_CONTAINER. REPACK_CONTTYPEID%TYPE;
56 v_CutoffUpdAllowed SIMULHOST_CONTAINER. CUTOFF_UPD_ALLOWED%TYPE;
57 v_DestCountry      SIMULHOST_CONTAINER. DESTINATION_COUNTRY%TYPE;
58
59 —FP
60 v_MasterFpName     SIMULHOST_FP. Master_Fp_Name%TYPE;
61 v_WsCount          SIMULHOST_FP. WS_CNT%TYPE := 0;
62 v_Status           SIMULHOST_FP. STATUS%TYPE;
63
64 —WS
65 v_WsNo             SIMULHOST_WS. WS_NO%TYPE := 1;
66 v_Priority         SIMULHOST_WS. Priority%TYPE := 0;
67 v_WsType           SIMULHOST_WS. WS_TYPE%TYPE;
68 v_LocNo            SIMULHOST_WS. LOCNO%TYPE;
69 v_WsQty            SIMULHOST_WS. Qty%TYPE := 0;
70 v_BoxQty           SIMULHOST_WS. BoxQty%TYPE;
71 v_ItemNo           SIMULHOST_WS. ItemNo%TYPE;
72 v_ItemDesc         SIMULHOST_WS. Item_Description%TYPE;
73 v_CntryOrigin      SIMULHOST_WS. COUNTRY_OF_ORIGIN%TYPE;
74 v_Asn              SIMULHOST_WS. ASN%TYPE;
75 v_PurchaseOrder    SIMULHOST_WS. PURCHASE_ORDER%TYPE;
76 v_ReturnStock      SIMULHOST_WS. RETURN_STOCK%TYPE;
77 v_VasCodeWs        SIMULHOST_WS. VAS_CODE%TYPE;
78 v_HostLineNo       SIMULHOST_WS. HOST_LINE_NO%TYPE;
79 BEGIN
80 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Start: ScenarioId='||p_ScenarioId||', Imp_Status='||p_ImpStatus);
81 — get unique EventId from the sequence
82 v_EventId := GetNewEvent_Id();
83 — get the scenario name and divide it into DAI Identifier, Action and SubAction
84 v_ScenarioType := GetParOneOfStrList('SCENARIO_TYPE', p_ScenarioId);
85 v_DaiIdentifier := REGEXP_SUBSTR(v_ScenarioType, '[^ ]+');
86 v_Action := SUBSTR(REGEXP_SUBSTR(v_ScenarioType, '\[[^ ]+\]', 2);
87 v_VarChar := REGEXP_SUBSTR(v_ScenarioType, '[^ ]+\)');
88 v_SubAction := SUBSTR(v_VarChar, 2, LENGTH(v_VarChar)-2);
89
90 — Initialize the needed variables according to DAI Identifier, Action and SubAction
91 IF (v_DaiIdentifier = EVENT_IDENTIFIER_IGD) THEN
92   IF (v_Action = EVENT_ACTION_ADD) THEN
93     IF (v_SubAction = EVENT_SUB_ACTION_CONTAINER) THEN
94       LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario IGD ADD CONTAINER');
95       — Event
96       v_Command := EVENT_COMMAND_ADD;
97       — Container
98       v_CntNo := CNTNO_IGD_ADD_CONTAINER;
99       v_StatusCont := CONT_STAT_IGD_ADD_CONTAINER;
100      v_WpTypeld := WP_TYPE_ID_PW;
101      — FP
102      v_MasterFpName := MFP_NAME_OSR;
103      v_Status := FP_STAT_IGD_ADD_CONTAINER;
104      — WS
105      v_WsType := WS_TYPE_ID_PUT;
106      FillTableMD_ART_TEMP(p_ScenarioId, FALSE);
```



```
170      VALUES(v_EventId, v_WpoNo, v_CntNo, v_MasterFpName, v_WsNo, v_Status, v_ErrorId, v_Priority,
171      v_WsType, v_LocNo, v_QtyReturnStockN, v_BoxQty, v_ItemNo, v_ItemDesc, p_ImpStatus,
172      gvc_ImpRemarkPfx||p_ScenarioId, v_CntryOrigin, v_Asn, v_PurchaseOrder, 'N', v_VasCodeWs, v_HostLineNo);
173
174      UPDATE SIMULHOST_FP SET WS_CNT=WS_CNT+1
175      WHERE EVENT_ID=v_EventId AND WPONO=v_WpoNo AND CNTNO=v_CntNo AND MASTER_FP_NAME=v_MasterFpName;
176  END IF;
177
178  IF (v_QtyReturnStockY != 0) THEN
179      LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_WS: EventId='||v_EventId
180      ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo||', MasterFP='||v_MasterFpName||', WsNo='||v_WsNo);
181      INSERT INTO SIMULHOST_WS(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_NO, STATUS, ERRORID, PRIORITY,
182      WS_TYPE, LOCNO, QTY, BOXQTY, ITEMNO, ITEM_DESCRIPTION, IMP_STATUS,
183      IMP_REMARK, COUNTRY_OF_ORIGIN, ASN, PURCHASE_ORDER, RETURN_STOCK, VAS_CODE, HOST_LINE_NO)
184      VALUES(v_EventId, v_WpoNo, v_CntNo, v_MasterFpName, v_WsNo, v_Status, v_ErrorId, v_Priority,
185      v_WsType, v_LocNo, v_QtyReturnStockY, v_BoxQty, v_ItemNo, v_ItemDesc, p_ImpStatus,
186      gvc_ImpRemarkPfx||p_ScenarioId, v_CntryOrigin, v_Asn, v_PurchaseOrder, 'Y', v_VasCodeWs, v_HostLineNo);
187
188      UPDATE SIMULHOST_FP SET WS_CNT=WS_CNT+1
189      WHERE EVENT_ID=v_EventId AND WPONO=v_WpoNo AND CNTNO=v_CntNo AND MASTER_FP_NAME=v_MasterFpName;
190  END IF;
191
192  END LOOP;
193  END LOOP;
194  DELETE FROM MD_ART_TEMP;
195  END IF;
196  END IF;
197  ELSIF(v_DaiIdentifier = EVENT_IDENTIFIERTMC) THEN
198  IF(v_Action = EVENT_ACTION_ADD) THEN
199  IF(v_SubAction = EVENT_SUB_ACTION_CONTAINER) THEN
200  LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario TMC ADD CONTAINER');
201  -- Event
202  v_Command := EVENT_COMMAND_EMPTY_TOTE;
203  -- Container
204  v_CntNo := CNTNO_TMC_ADD_CONTAINER;
205  v_StatusCont := CONT_STAT_TMC_ADD_CONTAINER;
206  v_PriorityCont := CONT_PRIO_TMC_ADD_CONTAINER;
207  v_WpTypeld := WP_TYPE_ID_SS;
208  v_ContTypeld := CONTTYPE_ID_TOT;
209  -- FP
210  v_MasterFpName := MFP_NAME_EMPTY;
211  v_Status := FP_STAT_TMC_ADD_CONTAINER;
212
213  --Write Event to database
214  LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
215  INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, COMMAND, DAI_IDENTIFIER, IMP_STATUS)
216  VALUES (v_EventId, v_Action, v_SubAction, v_Command, v_DaiIdentifier, p_ImpStatus);
217
218  vnu_ContainerCount := GetParOneOfRangeNum('ORDER_COUNT', p_ScenarioId, DFLT_PAR_INBOUND_CONT_COUNT);
219  LOG(vvc_Module, LOGLEVEL_DEBUG, TO_CHAR(vnu_ContainerCount)||' orders will be generated. ');
220  FOR vnu_ContainerIter IN 1..vnu_ContainerCount LOOP
221  v_WpoNo := SUBSTR(GetParOneOfStrList('ORDER_PREFIX', p_ScenarioId),1,10)||GetNewWpoNo();
222  v_Licenceplate := GetNewLicenceplateTOT();
223
224  LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_CONTAINER: EventId='||v_EventId
225  ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo);
226  INSERT INTO SIMULHOST_CONTAINER (Event_ID, WPONO, CNTNO, FP_CNT, STATUS, ERRORID, PRIORITY, WPTYPEID,
227  LICENCEPLATE, CONTTYPEID, IMP_STATUS, IMP_REMARK, VAS_CODE, CUTOFF_TIME, RETURN_REFERENCE, SHIP_VIA,
228  EXPORT_ORDER, CUSTOMER_ORDER_TIME, WMS_START_TIME, REPACK_CONTTYPEID, CUTOFF_UPD_ALLOWED, DESTINATION_COUNTRY)
229  VALUES (v_EventId, v_WpoNo, v_CntNo, v_FpCount, v_StatusCont, v_ErrorId, v_PriorityCont, v_WpTypeld,
230  v_Licenceplate, v_ContTypeld, p_ImpStatus, gvc_ImpRemarkPfx||p_ScenarioId, v_VasCodeCont, v_CutoffTime,
231  v_ReturnReference, v_ShipVia, v_ExportOrder, v_CustomerOrderTime, v_WmsStartTime, v_RepackContTypeld,
232  v_CutoffUpdAllowed, v_DestCountry);
```

```

233
234     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_FP: EventId='||v_EventId
235     ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo||', MasterFP='||v_MasterFpName);
236     INSERT INTO SIMULHOST_FP(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_CNT, STATUS, ERRORID, PRIORITY,
237     IMP_STATUS, IMP_REMARK, LICENCEPLATE)
238     VALUES (v_EventId, v_WpoNo, v_CntNo, v_MasterFpName, 0, v_Status, v_ErrorId, v_Priority,
239     p_ImpStatus, gvc_ImpRemarkPfx||p_ScenarioId, v_Licenceplate);
240     END LOOP;
241     END IF;
242     END IF;
243     ELSIF(v_DaiIdentifier = EVENT_IDENTIFIER_ORD) THEN
244     IF(v_Action = EVENT_ACTION_ADD) THEN
245     IF(v_SubAction = EVENT_SUB_ACTION_CONTAINER) THEN
246     LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario ORD ADD CONTAINER');
247     -- Event
248     v_Command := EVENT_COMMAND_NEW;
249     -- Container
250     v_StatusCont := CONT_STAT_ORD_ADD_CONTAINER;
251     v_WpTypeId := WP_TYPE_ID_CL;
252     v_ContTypeId := CONTTYPE_ID_TOT;
253     -- FP
254     v_MasterFpName := MFP_NAME_OSR;
255     v_Status := FP_STAT_ORD_ADD_CONTAINER;
256     -- WS
257     v_WsType := WS_TYPE_ID_GET;
258
259     --Write Event to database
260     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
261     INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, COMMAND, DAI_IDENTIFIER, IMP_STATUS)
262     VALUES (v_EventId, v_Action, v_SubAction, v_Command, v_DaiIdentifier, p_ImpStatus);
263
264     vnu_ContainerCount := GetParOneOfRangeNum('ORDER_COUNT', p_ScenarioId, DFLT_PAR_INBOUND_CONT_COUNT);
265     LOG(vvc_Module, LOGLEVEL_DEBUG, TO_CHAR(vnu_ContainerCount)||' orders will be generated. ');
266     DELETE FROM OSR_STOCK_TEMP;
267     INSERT INTO OSR_STOCK_TEMP(ITEMNO, ITEMDESC, QTY)
268     SELECT MD_ART.ITEMNO, MD_ART.ITEMDESC, V_OSR_STOCK.QTY
269     FROM MD_ART INNER JOIN V_OSR_STOCK
270     ON MD_ART.ITEMNO = V_OSR_STOCK.ITEMNO;
271     FillTableMD_ART_TEMP(P_ScenarioId, TRUE);
272
273     FOR vnu_ContainerIter IN 1..vnu_ContainerCount LOOP
274     v_WpoNo := SUBSTR(GetParOneOfStrList('ORDER_PREFIX', p_ScenarioId),1,10)||GetNewWpoNo();
275     v_CntNo := GetNewCntNo();
276     v_PriorityCont := GetParOneOfRangeNum('PRIORITY', p_ScenarioId, DFLT_PAR_COM_CONT_PRIORITY);
277     SELECT SYSDATE - INTERVAL '45' MINUTE INTO v_CustomerOrderTime FROM dual;
278     SELECT SYSDATE - INTERVAL '40' MINUTE INTO v_WmsStartTime FROM dual;
279     v_RepackContTypeId := GetParOneOfStrList('CONTTYPEID', p_ScenarioId, DFLT_PAR_COM_CONTTYPEID);
280     v_VasCodeCont := GetParOneOfStrList('VAS_CODE_CONT', p_ScenarioId);
281     v_ExportOrder := GetParOneOfStrList('EXPORT_ORDER', p_ScenarioId);
282     v_Offset := GetSingleOrRangeNum('CUTOFF_TIME', p_ScenarioId);
283     IF (v_Offset IS NOT NULL) THEN
284     SELECT SYSDATE + v_Offset/1440 INTO v_CutoffTime FROM dual;
285     END IF;
286     v_ShipVia := GetParOneOfStrList('SHIP_VIA', p_ScenarioId);
287     v_CutoffUpdAllowed := GetParOneOfStrList('CUTOFF_UPD_AL', p_ScenarioId);
288     v_DestCountry := GetParOneOfStrList('DEST_CNTRY', p_ScenarioId);
289
290     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_CONTAINER: EventId='||v_EventId
291     ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo);
292     INSERT INTO SIMULHOST_CONTAINER (Event_ID, WPONO, CNTNO, FP_CNT, STATUS, ERRORID, PRIORITY, WPTYPEID,
293     LICENCEPLATE, CONTTYPEID, IMP_STATUS, IMP_REMARK, VAS_CODE, CUTOFF_TIME, RETURN_REFERENCE, SHIP_VIA,
294     EXPORT_ORDER, CUSTOMER_ORDER_TIME, WMS_START_TIME, REPACK_CONTTYPEID, CUTOFF_UPD_ALLOWED, DESTINATION_COUNTRY)
295     VALUES (v_EventId, v_WpoNo, v_CntNo, v_FpCount, v_StatusCont, v_ErrorId, v_PriorityCont, v_WpTypeId,

```

```
296     v_Licenceplate, v_ContTypeld, p_ImpStatus, gvc_ImpRemarkPfx||p_ScenarioId, v_VasCodeCont, v_CutoffTime,
297     v_ReturnReference, v_ShipVia, v_ExportOrder, v_CustomerOrderTime, v_WmsStartTime, v_RepackContTypeld,
298     v_CutoffUpdAllowed, v_DestCountry);
299
300 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_FP: EventId='||v_EventId
301     ||', Wpono='||v_Wpono ||', CntNo='||v_CntNo||', MasterFP='||v_MasterFpName);
302 INSERT INTO SIMULHOST_FP(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_CNT, STATUS, ERRORID, PRIORITY,
303     IMP_STATUS, IMP_REMARK, LICENCEPLATE)
304     VALUES (v_EventId, v_Wpono, v_CntNo, v_MasterFpName, 0, v_Status, v_ErrorId, v_Priority,
305     p_ImpStatus, gvc_ImpRemarkPfx||p_ScenarioId, v_Licenceplate);
306
307 v_SinglePcsLines := GetParOneOfStrList('SINGLE_PCS_LNS', p_ScenarioId);
308 IF(v_SinglePcsLines = 'SP') THEN
309     vnu_WsCount := 1;
310     v_WsQty := 1;
311     setWsQty := FALSE;
312 ELSIF(v_SinglePcsLines = 'SL') THEN
313     vnu_WsCount := 1;
314     setWsQty := TRUE;
315 ELSIF(v_SinglePcsLines = 'R') THEN
316     vnu_WsCount := GetSingleOrRangeNum('QTY_ITEM', p_ScenarioId);
317     setWsQty := TRUE;
318 END IF;
319
320 LOG(vvc_Module, LOGLEVEL_DEBUG, TO_CHAR('Number of different SKUs = '||vnu_WsCount));
321 FOR vnu_WsIter IN 1..vnu_WsCount LOOP
322     v_WsNo := vnu_WsIter;
323     GetItemFromDb (p_ScenarioId, v_ItemNo, v_ItemDesc);
324     IF(setWsQty) THEN
325         v_WsQty := GetSingleOrRangeNum('QTY_PIECES_ITEM', p_ScenarioId);
326     END IF;
327     v_WsQty := DoStockcheck(p_ScenarioId, v_ItemNo, v_WsQty);
328     v_VasCodeWs := GetParOneOfStrList('VAS_CODE_WS', p_ScenarioId);
329     v_HostLineNo := GetNewHostLineNo();
330
331     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_WS: EventId='||v_EventId
332         ||', Wpono='||v_Wpono ||', CntNo='||v_CntNo||', MasterFP='||v_MasterFpName||', WsNo='||v_WsNo);
333     INSERT INTO SIMULHOST_WS(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_NO, STATUS, ERRORID, PRIORITY,
334         WS_TYPE, LOCNO, QTY, BOXQTY, ITEMNO, ITEM_DESCRIPTION, IMP_STATUS,
335         IMP_REMARK, COUNTRY_OF_ORIGIN, ASN, PURCHASE_ORDER, RETURN_STOCK, VAS_CODE, HOST_LINE_NO)
336         VALUES(v_EventId, v_Wpono, v_CntNo, v_MasterFpName, v_WsNo, v_Status, v_ErrorId, v_Priority,
337         v_WsType, v_LocNo, v_WsQty, v_BoxQty, v_ItemNo, v_ItemDesc, p_ImpStatus,
338         gvc_ImpRemarkPfx||p_ScenarioId, v_CntryOrigin, v_Asn, v_PurchaseOrder, v_ReturnStock,
339         v_VasCodeWs, v_HostLineNo);
340
341     UPDATE SIMULHOST_FP SET WS_CNT=WS_CNT+1
342         WHERE EVENT_ID=v_EventId AND WPONO=v_Wpono AND CNTNO=v_CntNo AND MASTER_FP_NAME=v_MasterFpName;
343
344     END LOOP;
345 END LOOP;
346 DELETE FROM MD_ART_TEMP;
347 END IF;
348 END IF;
349 ELSIF(v_DaiIdentifier = EVENT_IDENTIFIER_STR) THEN
350     IF(v_Action = EVENT_ACTION_ADD) THEN
351         IF(v_SubAction = EVENT_SUB_ACTION_CONTAINER) THEN
352             v_StrSeperator := SUBSTR(REGEXP_SUBSTR(v_ScenarioType, '- [^ ]+', 3));
353             -- Event
354             IF(v_StrSeperator = 'REPL') THEN
355                 LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario STR ADD CONTAINER - REPL');
356                 v_Command := EVENT_COMMAND_DEMAND;
357                 v_WpTypeld := WP_TYPE_ID_RE;
358                 isStockcheck := TRUE;
```



```
359     ELSIF(v_StrSeperator = 'EOS') THEN
360         LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario STR ADD CONTAINER – EOS');
361         v_Command := EVENT_COMMAND_DESLOT;
362         v_WpTypeld := WP_TYPE_ID_HK;
363         isStockcheck := FALSE;
364     END IF;
365     — Container
366     v_CntNo := CNTNO_IGD_ADD_CONTAINER;
367     v_FpCount := 2;
368     v_StatusCont := CONT_STAT_ORD_ADD_CONTAINER;
369     v_PriorityCont := 0;
370     v_ContTypeld := CONTTYPE_ID_TOT;
371     — FP
372     v_Status := FP_STAT_ORD_ADD_CONTAINER;
373     — WS
374     v_WsType := WS_TYPE_ID_GET;
375
376     —Write Event to database
377     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
378     INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, COMMAND, DAI_IDENTIFIER, IMP_STATUS)
379     VALUES (v_EventId, v_Action, v_SubAction, v_Command, v_DaiIdentifier, p_ImpStatus);
380
381     vnu_ContainerCount := GetParOneOfRangeNum('ORDER_COUNT', p_ScenarioId, DFLT_PAR_INBOUND_CONT_COUNT);
382     LOG(vvc_Module, LOGLEVEL_DEBUG, TO_CHAR(vnu_ContainerCount)||' orders will be generated. ');
383
384     INSERT INTO OSR_STOCK_TEMP(ITEMNO, ITEMDESC, QTY)
385     SELECT MD_ART.ITEMNO, MD_ART.ITEMDESC, V_OSR_STOCK.QTY
386     FROM MD_ART INNER JOIN V_OSR_STOCK
387     ON MD_ART.ITEMNO = V_OSR_STOCK.ITEMNO;
388     FillTableMD_ART_TEMP(p_ScenarioId, isStockcheck);
389
390     FOR vnu_ContainerIter IN 1..vnu_ContainerCount LOOP
391         v_WpoNo := SUBSTR(GetParOneOfStrList('ORDER_PREFIX', p_ScenarioId),1,10)||GetNewWpoNo();
392         v_ReturnReference := v_WpoNo;
393
394         LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_CONTAINER: EventId='||v_EventId
395         ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo);
396         INSERT INTO SIMULHOST_CONTAINER (Event_ID, WPONO, CNTNO, FP_CNT, STATUS, ERRORID, PRIORITY, WPTYPEID,
397         LICENCEPLATE, CONTTYPEID, IMP_STATUS, IMP_REMARK, VAS_CODE, CUTOFF_TIME, RETURN_REFERENCE, SHIP_VIA,
398         EXPORT_ORDER, CUSTOMER_ORDER_TIME, WMS_START_TIME, REPACK_CONTTYPEID, CUTOFF_UPD_ALLOWED, DESTINATION_COUNTRY)
399         VALUES (v_EventId, v_WpoNo, v_CntNo, v_FpCount, v_StatusCont, v_ErrorId, v_PriorityCont, v_WpTypeld,
400         v_Licenceplate, v_ContTypeld, p_ImpStatus, gvc_ImpRemarkPfx||p_ScenarioId, v_VasCodeCont, v_CutoffTime,
401         v_ReturnReference, v_ShipVia, v_ExportOrder, v_CustomerOrderTime, v_WmsStartTime, v_RepackContTypeld,
402         v_CutoffUpdAllowed, v_DestCountry);
403
404         v_MasterFpName := MFP_NAME_OSR;
405
406         LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_FP: EventId='||v_EventId
407         ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo||', MasterFP='||v_MasterFpName);
408         INSERT INTO SIMULHOST_FP(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_CNT, STATUS, ERRORID, PRIORITY,
409         IMP_STATUS, IMP_REMARK, LICENCEPLATE)
410         VALUES (v_EventId, v_WpoNo, v_CntNo, v_MasterFpName, 0, v_Status, v_ErrorId, v_Priority,
411         p_ImpStatus, gvc_ImpRemarkPfx||p_ScenarioId, v_Licenceplate);
412
413         vnu_WsCount := GetSingleOrRangeNum('QTY_ITEM', p_ScenarioId);
414         If vnu_WsCount = 0 THEN
415             SELECT COUNT(*) INTO v_RowCount FROM MD_ART_TEMP;
416             IF(v_RowCount > 0) THEN
417                 vnu_WsCount := ROUND(DBMS_RANDOM.VALUE(1,v_RowCount));
418             ELSE
419                 vnu_WsCount := 0;
420             End IF;
421         END IF;
```

```

422 LOG(vvc_Module, LOGLEVEL_DEBUG, TO_CHAR('Number of different SKUs = '||vnu_WsCount));
423 FOR vnu_WsIter IN 1..vnu_WsCount LOOP
424     v_WsNo := vnu_WsIter;
425     GetItemFromDb (p_ScenarioId, v_ItemNo, v_ItemDesc);
426     IF(v_StrSeperator = 'REPL' AND v_ItemNo != 'No Data Found') THEN
427         v_Stockcheck := GetParOneOfStrList('STOCKCHECK', p_ScenarioId);
428         IF(UPPER(v_Stockcheck) = 'TRUE') THEN
429             SELECT QTY INTO v_CurrentStock FROM OSR_STOCK_TEMP
430                 WHERE ITEMNO = v_ItemNo;
431             IF(v_CurrentStock IS NOT NULL) THEN
432                 v_WsQty := ROUND(DBMS_RANDOM.VALUE(1,v_CurrentStock));
433                 IF(v_WsQty = v_CurrentStock) THEN
434                     -- Delete from stock table
435                     DELETE FROM OSR_STOCK_TEMP
436                         WHERE ITEMNO = v_ItemNo;
437                 ELSE
438                     -- Update stock table
439                     UPDATE OSR_STOCK_TEMP
440                         SET QTY = QTY - v_WsQty
441                         WHERE ITEMNO = v_ItemNo;
442                 END IF;
443             END IF;
444         ELSE
445             v_WsQty := GetSingleOrRangeNum('QTY_PIECES_ITEM', p_ScenarioId);
446         END IF;
447     ELSIF(v_StrSeperator = 'EOS') THEN
448         v_WsQty := 0;
449     END IF;
450
451     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_WS: EventId='||v_EventId
452         ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo||', MasterFP='||v_MasterFpName||', WsNo='||v_WsNo);
453     INSERT INTO SIMULHOST_WS(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_NO, STATUS, ERRORID, PRIORITY,
454     WS_TYPE, LOCNO, QTY, BOXQTY, ITEMNO, ITEM_DESCRIPTION, IMP_STATUS,
455     IMP_REMARK, COUNTRY_OF_ORIGIN, ASN, PURCHASE_ORDER, RETURN_STOCK, VAS_CODE, HOST_LINE_NO)
456     VALUES(v_EventId, v_WpoNo, v_CntNo, v_MasterFpName, v_WsNo, v_Status, v_ErrorId, v_Priority,
457     v_WsType, v_LocNo, v_WsQty, v_BoxQty, v_ItemNo, v_ItemDesc, p_ImpStatus,
458     gvc_ImpRemarkPfx||p_ScenarioId, v_CntryOrigin, v_Asn, v_PurchaseOrder, v_ReturnStock,
459     v_VasCodeWs, v_HostLineNo);
460
461     UPDATE SIMULHOST_FP SET WS_CNT=WS_CNT+1
462     WHERE EVENT_ID=v_EventId AND WPONO=v_WpoNo AND CNTNO=v_CntNo AND MASTER_FP_NAME=v_MasterFpName;
463 END LOOP;
464
465 v_MasterFpName := MFP_NAME_CDC;
466
467 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_FP: EventId='||v_EventId
468     ||', WpoNo='||v_WpoNo ||', CntNo='||v_CntNo||', MasterFP='||v_MasterFpName);
469     INSERT INTO SIMULHOST_FP(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_CNT, STATUS, ERRORID, PRIORITY,
470     IMP_STATUS, IMP_REMARK, LICENCEPLATE)
471     VALUES (v_EventId, v_WpoNo, v_CntNo, v_MasterFpName, 0, v_Status, v_ErrorId, v_Priority,
472     p_ImpStatus, gvc_ImpRemarkPfx||p_ScenarioId, v_Licenceplate);
473 END LOOP;
474 DELETE FROM MD_ART_TEMP;
475 END IF;
476 END IF;
477 END IF;
478
479 UPDATE ACT_SCENARIO
480     SET DLE = SYSDATE
481     WHERE SCENARIO_ID = p_ScenarioId;
482 COMMIT;
483 pnu_RetCode := GLOBAL_RETURN_CODE_OK;
484 EXCEPTION

```

```
485 WHEN OTHERS THEN
486     ROLLBACK;
487     LOG(vvc_Module, LOGLEVEL_ERROR, 'SQLERRM='||SQLERRM);
488     pnu_RetCode := GLOBAL_RETURN_CODE_ERROR;
489 END GenerateOrders; — procedure
490
491
492 /*
493 <type>         proc
494 <name>         ModContainer
495 <short>       Generates the entries for ORD MOD CONTAINER Scenario
496 <author>      Herka Jakob
497 */
498 PROCEDURE ModContainer (pnu_RetCode      OUT NUMBER,
499                        p_EventId        IN SIMULHOST_CONTAINER.EVENT_ID%TYPE,
500                        p_WpoNo          IN SIMULHOST_CONTAINER.WPONO%TYPE,
501                        p_CntNo          IN SIMULHOST_CONTAINER.CNTNO%TYPE,
502                        p_Priority        IN SIMULHOST_CONTAINER.PRIORITY%TYPE,
503                        p_CutoffTime      IN SIMULHOST_CONTAINER.CUTOFF_TIME%TYPE,
504                        p_ShipVia         IN SIMULHOST_CONTAINER.SHIP_VIA%TYPE,
505                        p_ForwardToWcs    IN NVARCHAR2)
506 IS
507     vvc_Module LOG_ENTRIES.SENDER%TYPE := gvc_ModuleHeader||'ModContainer';
508     v_EventId  SIMULHOST_Event.ID%TYPE;
509 BEGIN
510     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Start: EventId='||p_EventId||', WpoNo='||p_WpoNo
511     ||', CntNo='||p_CntNo||', Priority='||p_Priority||', CutoffTime='||p_CutoffTime
512     ||', ShipVia='||p_ShipVia||', ForwardToWcs='||p_ForwardToWcs);
513     LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario ORD MOD CONTAINER');
514     v_EventId := GetNewEvent_Id();
515     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
516     INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, COMMAND, DAI_IDENTIFIER, IMP_STATUS)
517     VALUES (v_EventId, EVENT_ACTION_MOD, EVENT_SUB_ACTION_CONTAINER, EVENT_COMMAND_AMEND,
518     EVENT_IDENTIFIER_ORD, GENERIC_IMP_STAT_GENERATED);
519
520     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_CONTAINER: EventId='||v_EventId
521     ||', WpoNo='||p_WpoNo||', CntNo='||p_CntNo);
522     INSERT INTO SIMULHOST_CONTAINER (EVENT_ID, WPONO, CNTNO, FP_CNT, STATUS, ERRORID, PRIORITY, WPTYPEID,
523     LICENCEPLATE, CONTTYPEID, IMP_STATUS, IMP_REMARK, VAS_CODE, CUTOFF_TIME, RETURN_REFERENCE, SHIP_VIA,
524     EXPORT_ORDER, CUSTOMER_ORDER_TIME, WMS_START_TIME, REPACK_CONTTYPEID, CUTOFF_UPD_ALLOWED, DESTINATION_COUNTRY)
525     SELECT v_EventId, p_WpoNo, p_Cntno, 0, CONST.CONT_STAT_MARRIED, ERRORID, p_Priority, WPTYPEID,
526     LICENCEPLATE, CONTTYPEID, IMP_STATUS, IMP_REMARK, VAS_CODE, p_CutoffTime, RETURN_REFERENCE, p_ShipVia,
527     EXPORT_ORDER, CUSTOMER_ORDER_TIME, WMS_START_TIME, REPACK_CONTTYPEID, 'N', DESTINATION_COUNTRY
528     FROM SIMULHOST_CONTAINER
529     WHERE EVENT_ID = p_EventId AND WPONO = p_WpoNo AND CNTNO = p_CntNo;
530
531     COMMIT;
532     pnu_RetCode := CheckForwardToWcs(v_EventId, p_ForwardToWcs);
533 EXCEPTION
534 WHEN OTHERS THEN
535     ROLLBACK;
536     LOG(vvc_Module, LOGLEVEL_ERROR, 'SQLERRM='||SQLERRM);
537     pnu_RetCode := GLOBAL_RETURN_CODE_ERROR;
538 END ModContainer; — procedure
539
540
541 /*
542 <type>         proc
543 <name>         DelContainer
544 <short>       Generates the entries for a DEL CONTAINER Scenario
545 <author>      Herka Jakob
546 */
547 PROCEDURE DelContainer (pnu_RetCode      OUT NUMBER,
```

```
548         p_EventId      IN SIMULHOST_CONTAINER.EVENT_ID%TYPE,
549         p_WpoNo        IN SIMULHOST_CONTAINER.WPONO%TYPE,
550         p_CntNo        IN SIMULHOST_CONTAINER.CNTNO%TYPE,
551         p_ForwardToWcs IN NVARCHAR2,
552         p_DaiIdentifier IN SIMULHOST_EVENT.DAI_IDENTIFIER%TYPE,
553         p_Command      IN SIMULHOST_EVENT.COMMAND%TYPE,
554         p_WpTypeId     IN SIMULHOST_CONTAINER.WPTYPEID%TYPE := NULL)
555 IS
556     vvc_Module LOG_ENTRIES.SENDER%TYPE := gvc_ModuleHeader||'DelContainer';
557     v_EventId  SIMULHOST_Event.ID%TYPE;
558 BEGIN
559     v_EventId := GetNewEvent_Id();
560     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
561     INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, COMMAND, DAI_IDENTIFIER, IMP_STATUS)
562     VALUES (v_EventId, EVENT_ACTION_DEL, EVENT_SUB_ACTION_CONTAINER, p_Command, p_DaiIdentifier,
563     GENERIC_IMP_STAT_GENERATED);
564
565     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_CONTAINER: EventId='||v_EventId
566     ||', WpoNo='||p_WpoNo ||', CntNo='||p_CntNo);
567     INSERT INTO SIMULHOST_CONTAINER (EVENT_ID, WPONO, CNTNO, FP_CNT, STATUS, ERRORID, PRIORITY, WPTYPEID,
568     LICENCEPLATE, CONTTYPEID, IMP_STATUS, IMP_REMARK, VAS_CODE, CUTOFF_TIME, RETURN_REFERENCE, SHIP_VIA,
569     EXPORT_ORDER, CUSTOMER_ORDER_TIME, WMS_START_TIME, REPACK_CONTTYPEID, CUTOFF_UPD_ALLOWED, DESTINATION_COUNTRY)
570     SELECT v_EventId, p_WpoNo, p_Cntno, 0, CONT_STAT_DEL_CONTAINER, ERRORID, PRIORITY, NVL(p_WpTypeId, WPTYPEID),
571     LICENCEPLATE, CONTTYPEID, IMP_STATUS, IMP_REMARK, VAS_CODE, CUTOFF_TIME, RETURN_REFERENCE, SHIP_VIA,
572     EXPORT_ORDER, CUSTOMER_ORDER_TIME, WMS_START_TIME, REPACK_CONTTYPEID, CUTOFF_UPD_ALLOWED, DESTINATION_COUNTRY
573     FROM SIMULHOST_CONTAINER
574     WHERE EVENT_ID = p_EventId AND WPONO = p_WpoNo AND CNTNO = p_CntNo;
575
576     COMMIT;
577     pnu_RetCode := CheckForwardToWcs(v_EventId, p_ForwardToWcs);
578 EXCEPTION
579     WHEN OTHERS THEN
580         ROLLBACK;
581         LOG(vvc_Module, LOGLEVEL_ERROR, 'SQLERRM='||SQLERRM);
582         pnu_RetCode := GLOBAL_RETURN_CODE_ERROR;
583 END DelContainer; --procedure
584
585
586 /*
587 <type>      proc
588 <name>      AddFp
589 <short>     Generates the entries for TMR ADD FP Scenario
590 <author>    Herka Jakob
591 */
592 PROCEDURE AddFp (pnu_RetCode OUT NUMBER,
593                 p_WpoNo     IN SIMULHOST_CONTAINER.WPONO%TYPE,
594                 p_CntNo     IN SIMULHOST_CONTAINER.CNTNO%TYPE,
595                 p_Licenceplate IN SIMULHOST_CONTAINER.LICENCEPLATE%TYPE,
596                 p_MasterFpName IN SIMULHOST_FP.MASTER_FP_NAME%TYPE,
597                 p_ForwardToWcs IN NVARCHAR2,
598                 p_Command    IN SIMULHOST_EVENT.COMMAND%TYPE)
599 IS
600     vvc_Module LOG_ENTRIES.SENDER%TYPE := gvc_ModuleHeader||'AddFp';
601     v_EventId  SIMULHOST_Event.ID%TYPE;
602     v_ImpStatus SIMULHOST_Event.Imp_Status%TYPE := GENERIC_IMP_STAT_GENERATED;
603     v_MasterFpName SIMULHOST_FP.MASTER_FP_NAME%TYPE;
604 BEGIN
605     IF (p_MasterFpName IS NULL) THEN
606         SELECT NAME INTO v_MasterFpName FROM
607         (SELECT * FROM V_DSP_MASTER_FP
608         ORDER BY DBMS_RANDOM.VALUE)
609         WHERE ROWNUM = 1;
610     ELSE
```

```
611     v_MasterFpName := p_MasterFpName;
612 END IF;
613 v_EventId := GetNewEvent_Id();
614 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
615 INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, COMMAND, DAI_IDENTIFIER, IMP_STATUS)
616     VALUES (v_EventId, EVENT_ACTION_ADD, EVENT_SUB_ACTION_FP, p_Command, EVENT_IDENTIFIER_TMR, v_ImpStatus);
617
618 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_FP: EventId='||v_EventId
619     ||', WpoNo='||p_WpoNo ||', CntNo='||p_CntNo||', MasterFP='||p_MasterFpName);
620 INSERT INTO SIMULHOST_FP(EVENT_ID, WPONO, CNTNO, MASTER_FP_NAME, WS_CNT, STATUS, IMP_STATUS, LICENCEPLATE)
621     VALUES (v_EventId, p_WpoNo, p_CntNo, v_MasterFpName, 0, FP_STAT_JGD_ADD_CONTAINER, v_ImpStatus, p_Licenceplate);
622
623 pnu_RetCode := CheckForwardToWcs(v_EventId, p_ForwardToWcs);
624 EXCEPTION
625     WHEN OTHERS THEN
626         pnu_RetCode := GLOBAL_RETURN_CODE_ERROR;
627         LOG(vvc_Module, LOGLEVEL_ERROR, 'SQLERRM='||SQLERRM);
628 END AddFp; ---procedure
629
630
631 /*
632 <type>         proc
633 <name>         AddArticle
634 <short>       Generates the entries for PDU ADD ARTICLE Scenario
635 <author>      Herka Jakob
636 */
637 PROCEDURE AddArticle (pnu_RetCode      OUT NUMBER,
638                     p_ForwardToWcs    IN NVARCHAR2)
639 IS
640     vvc_Module LOG_ENTRIES.SENDER%TYPE := gvc_ModuleHeader||'AddArticle';
641     v_EventId  SIMULHOST_Event.ID%TYPE;
642 BEGIN
643     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Start: ForwardToWcs='||p_ForwardToWcs);
644     LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario PDU ADD ARTICLE');
645
646     v_EventId := GetNewEvent_Id();
647     LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
648     INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, DAI_IDENTIFIER, IMP_STATUS)
649         VALUES (v_EventId, EVENT_ACTION_ADD, EVENT_SUB_ACTION_ARTICLE, EVENT_IDENTIFIER_PDU,
650             GENERIC_IMP_STAT_GENERATED);
651     pnu_RetCode := CheckForwardToWcs(v_EventId, p_ForwardToWcs);
652 EXCEPTION
653     WHEN OTHERS THEN
654         ROLLBACK;
655         pnu_RetCode := GLOBAL_RETURN_CODE_ERROR;
656         LOG(vvc_Module, LOGLEVEL_ERROR, 'SQLERRM='||SQLERRM);
657 END AddArticle; ---procedure
658
659
660 /*
661 <type>         proc
662 <name>         ModStock
663 <short>       Generates the entries for SST MOD STOCK Scenario
664 <author>      Herka Jakob
665 */
666 PROCEDURE ModStock (pnu_RetCode      OUT NUMBER,
667                   p_ItemNo          IN SIMULHOST_STOCK.ITEMNO%TYPE,
668                   p_StatusNew       IN SIMULHOST_STOCK.STATUS_NEW%TYPE,
669                   p_ForwardToWcs    IN NVARCHAR2)
670 IS
671     vvc_Module LOG_ENTRIES.SENDER%TYPE := gvc_ModuleHeader||'ModStock';
672     v_EventId  SIMULHOST_Event.ID%TYPE;
673 BEGIN
```

```
674 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Start: ItemNo='||p_ItemNo ||', StatusNew='||p_StatusNew
675 ||', ForwardToWcs='||p_ForwardToWcs);
676 LOG(vvc_Module, LOGLEVEL_INFO, 'Create entries for scenario SST MOD STOCK');
677
678 v_EventId := GetNewEvent_Id();
679 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_EVENT: EventId='||v_EventId);
680 INSERT INTO SIMULHOST_EVENT (ID, ACTION, SUB_ACTION, DAI_IDENTIFIER, IMP_STATUS)
681 VALUES (v_EventId, EVENT_ACTION_MOD, EVENT_SUB_ACTION_STOCK, EVENT_IDENTIFIER_SST, GENERIC_IMP_STAT_GENERATED);
682
683 LOG(vvc_Module, LOGLEVEL_DEBUG, 'Create entry in SIMULHOST_STOCK: EventId='||v_EventId
684 ||', ItemNo='||p_ItemNo);
685 INSERT INTO SIMULHOST_STOCK(EVENT_ID, ITEMNO, STATUS_NEW)
686 VALUES (v_EventId, p_ItemNo, p_StatusNew);
687
688 COMMIT;
689 pnu_RetCode := CheckForwardToWcs(v_EventId, p_ForwardToWcs);
690 EXCEPTION
691 WHEN OTHERS THEN
692 ROLLBACK;
693 pnu_RetCode := GLOBAL_RETURN_CODE_ERROR;
694 LOG(vvc_Module, LOGLEVEL_ERROR, 'SQLERRM='||SQLERRM);
695 END ModStock; -- procedure
```