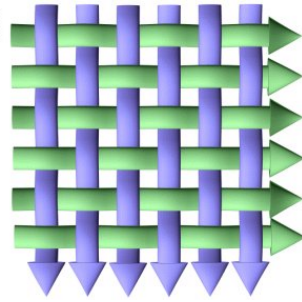


Sonderforschungsbereich 559
**Modellierung großer
Netze in der Logistik**



Technical Report 03016
ISSN 1612-1376

Beispielmodellierung eines Güterverkehrszentrums im ProC/B-Paradigma

Teilprojekt A4:
Christian Dilling

Teilprojekt M1:
Marcus Völker

Universität Dortmund
Lehrstuhl für Förder- und Lagerwesen
D-44221 Dortmund

Universität Dortmund
Lehrstuhl Informatik IV
D-44221 Dortmund

Dortmund, 22. August 2003

1 Einleitung

Der Güterverkehrsmarkt in Europa ist einem strukturellen Wandel unterworfen. In den vergangenen Jahren ist eine starke Zunahme des Transportaufwandes zu verzeichnen, die sich in steigenden Lieferfrequenzen bei einer gleichzeitigen Reduktion der Sendungsgrößen widerspiegelt. Die Gründe für diese Entwicklung sind im Wandel von einer Vorratswirtschaft zu einer Bedarfswirtschaft zu finden. Die Umsetzung von Just-in-time-Konzepten, die Einflüsse des E-Commerce, aber auch die wachsende industrielle Arbeitsteiligkeit und der Güterstruktureffekt haben in diesem Zusammenhang zu einem erhöhten Verkehrsaufkommen insbesondere auf der Straße geführt, da der LKW aufgrund seiner Flexibilität für viele Logistikdienstleister und Transportunternehmer das Hauptverkehrsmittel darstellt.

Um dem drohenden „Verkehrsinfarkt“ entgegenzuwirken und ökologischen Aspekten gerecht zu werden sind alternative Verkehrskonzepte zur Verkehrsvermeidung und -reduzierung gefordert. Güterverkehrszentren (GVZ) sind logistische Knotenpunkte, die dieser Forderung durch die Bündelung von Sendungen und dem Einbeziehen aller Verkehrsträger nachkommen. Darüber hinaus können Synergieeffekte durch Kooperation erzielt und insbesondere im Transportwesen enorme Economies of scale (Größenvorteile) realisiert werden.

In der Praxis hingegen mangelt es an der Umsetzung des originären GVZ-Gedankens. Zum einen fehlen die wissenschaftlichen Grundlagen bei der Planung von GVZ, zum anderen werden GVZ oft unternehmerisch motiviert von einem eng begrenzten Konsortium von Industrieunternehmen gegründet, wobei die mangelnde Kooperationsbereitschaft die Einbindung weiterer Unternehmen verhindert und damit dem Leitgedanken eines GVZ von der Kooperation möglichst vieler Logistikdienstleister widerspricht. Es fehlt an der wissenschaftlichen Differenzierung und Vernetzung der einzelnen Prozesse, Dienstleister und Transportketten, um das Optimum des Gesamtsystems GVZ zu erreichen. Um die Komplexität der Strukturen und Prozesse abbilden und systematisch analysieren zu können, ist im SFB 559 ein GVZ modelliert worden. Mit Hilfe dieses praxisgetreuen Modells sollen Gestaltungsrichtlinien für neu zu planende GVZ gewonnen und Verbesserungspotentiale für bereits bestehende GVZ abgeleitet werden.

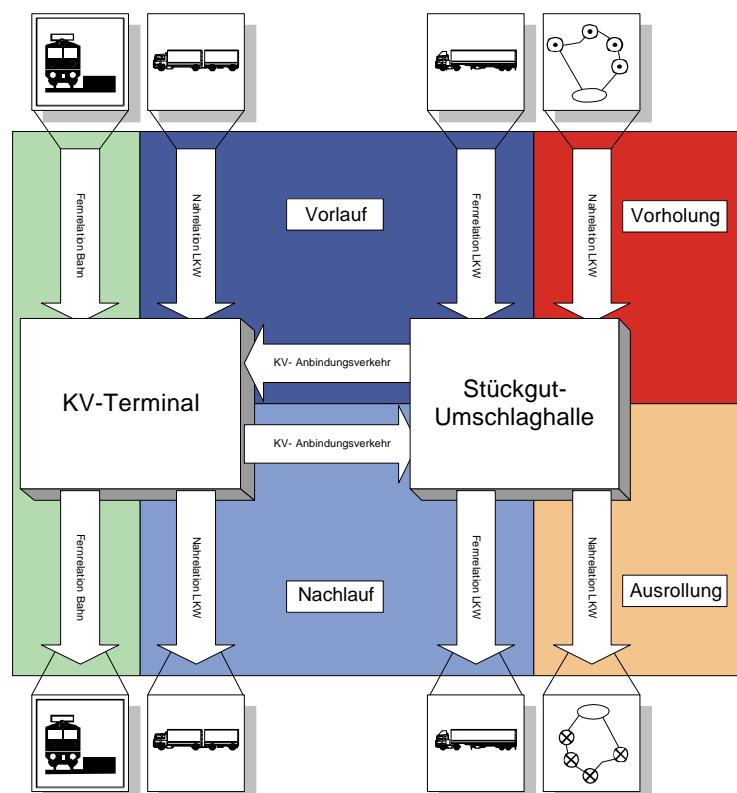


Abb. 1: Grundgestalt eines GVZ.

2 Modellbildung

2.1 Modellierungsobjekt

Ein Güterverkehrszentrum ist ein logistischer Knoten, in dem Stückgutströme aus Sammelfahrten ankommen, in einzelne Leistungsobjekte aufgebrochen, gelagert, konsolidiert, umgeschlagen und dann in Verteilfahrten ausgeliefert werden, wobei nicht notwendigerweise alle Vorgänge an einem Leistungsobjekt durchgeführt werden müssen. Ein GVZ setzt sich aus verschiedenen Funktionsbereichen bzw. Modulen zusammen. Die GVZ-Grundgestalt bilden die Module Stückgutumschlaghalle und KV-Terminal, sowie ergänzende Dienstleistungseinrichtungen, wobei die fundamentalen und für die Leistungsbetrachtung eines GVZ relevanten Prozesse wie Umschlag und Lagerung von Leistungsobjekten in den Modulen Stückgutumschlaghalle und KV-Terminal stattfinden (vgl. Abb. 1).

Ein Güterverkehrszentrum besteht somit aus verschiedenen Subsystemen, die in dynamischen Beziehungen miteinander interagieren. Die einzelnen Subsysteme greifen wiederum auf unterschiedliche Ressourcen zu.

2.2 Modellierungstool

Ein Modell ist eine vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System. Es unterscheidet sich hinsichtlich der untersuchungsrelevanten Eigenschaften nur innerhalb eines vom Untersuchungsziel abhängigen Toleranzrahmens vom Vorbild [VDI00]. Diese Definition trifft auf das entwickelte Modell eines GVZ mit seinen integrierten Modulen zu.

Für die Betrachtung der komplexen Strukturen eines GVZ eignet sich insbesondere das im SFB entwickelte ProC/B-Paradigma. Dieses ermöglicht die Abbildung von logistischen Prozessen und die Hinterlegung von Steuerungsstrategien. Darüber hinaus verbindet das ProC/B-Toolset diese Funktionalitäten in einer Softwarelösung und wurde daher zur Modellierung des GVZ herangezogen.

2.3 Umsetzung

In einem ersten Schritt zur Modellerstellung wurden die im GVZ ablaufenden Prozesse aufgenommen und in Form von anschaulichen Flussdiagrammen dargestellt. Diese wurden dann der Prozesskettentheorie entsprechend umgeformt und in Prozesskettenpläne überführt. Zur Umsetzung in ein integriertes Modell wurde das GVZ zunächst in Module aufgeteilt und mit einer verbindenden Steuerungsebene versehen. Die einzelnen Module sind hierarchisch in detailliertere Subsysteme untergliedert, in denen jeweils die eigentlichen Prozessketten hinterlegt sind. Dieser hierarchische und modulare Ansatz gewährleistet Flexibilität und Erweiterbarkeit, indem ganze Module oder einzelne Subsysteme ergänzt bzw. an Vorgaben aus der Praxis angepasst werden können.

2.4 Modell des GVZ

2.4.1 Grundlagen des ProC/B-Modellierungsparadigmas

Das ProC/B-Paradigma [BBF+02] ist eine Beschreibungsmöglichkeit, die auf dem Prozesskettenparadigma nach Kuhn [Ku95] beruht. Die Beschreibung umfasst dabei sowohl die Struktur als auch das Verhalten des zu modellierenden Systems.

Entsprechend der Gliederung realer Systeme bietet ProC/B **Funktionseinheiten** (FEs), die nach außen an definierten Schnittstellen Dienste anbieten und wiederum Dienste anderer Funktionseinheiten verwenden können, um ihre Aufgaben zu erfüllen. Die Dienste werden durch Prozessketten beschrieben, die durch Prozesskettenelemente (PKEs) eine Aktivitätenfolge modellieren. PKEs können dabei Verzögerungen, Dienstaufrufe oder Hi-Slang-Code [HIT93] enthalten.

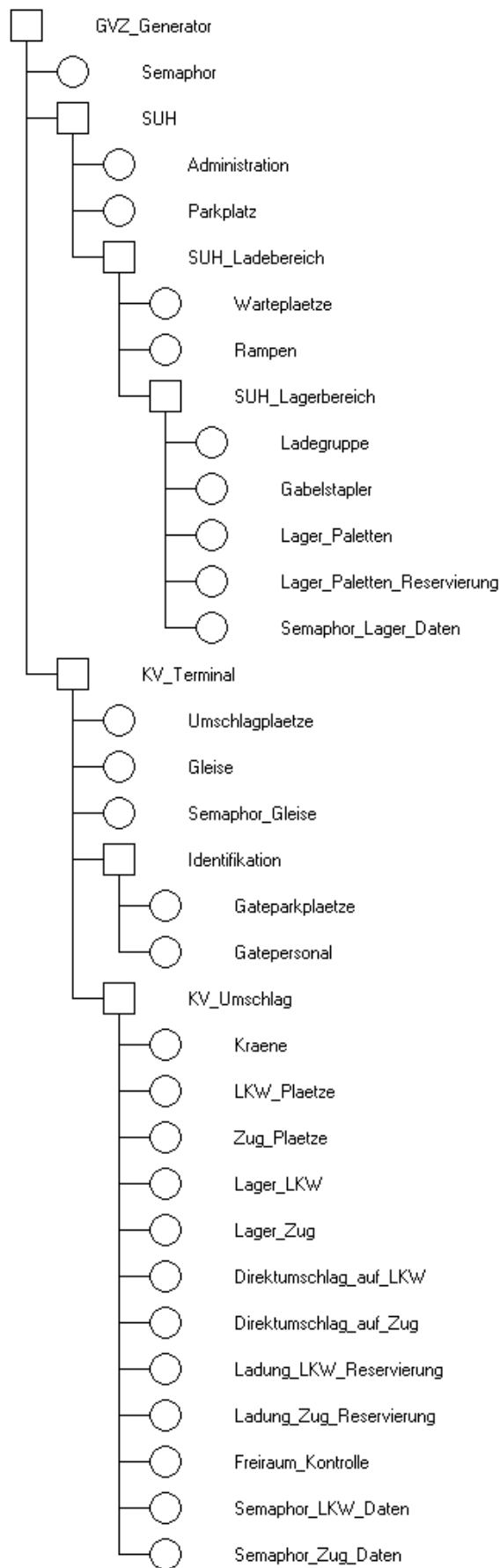


Abb. 2: Hierarchie des Modells.

Nach unten abgeschlossen wird die auf diese Weise entstehende FE-Hierarchie durch standardisierte FEs (Server, Counter, Storages). Ein **Server** dient zur Modellierung eines Zeitverbrauchs an Ressourcen. Die Serverleistung kann für eine bestimmte Zeit angefordert werden, und eignet sich zur Abbildung von Arbeitsmitteln oder Personal. Ein **Counter** stellt einen Raumverbrauch dar und eignet sich zur Modellierung eines Puffers oder einer begrenzten Menge an Arbeitsmitteln. Ein **Storage** bietet gegenüber einem Counter eine einfachere Handhabung der Modellierung bei der gemeinsamen Verwaltung mehrerer Lagergüter sowie eine Zustandsfunktion. Counter und Storages lassen sich in den Werten Ober- und Untergrenze sowie Initialwert variieren. Im GVZ zählen unter anderen die Abstellspuren, Gleise, Rampen, Relationsflächen, aber auch die Krane und Gabelstapler zu den Einheiten, die durch Storages modelliert werden.

Nach oben abgeschlossen wird die FE-Hierarchie durch **Quellen** und **Senken**, die eine Schnittstelle zur Umwelt darstellen. Die Quellen werden dazu analog des an den Schnittstellen beobachtbaren Eingangsverhalten parametrisiert.

Zur Darstellung paralleler oder alternativer Abläufe verfügt ProC/B über Konnektoren.

2.4.2 Allgemeine Beschreibung des Modells

Die **Struktur** des Modells (vgl. Abb. 2) besteht aus einer übergeordneten Steuerungsebene und den beiden hierarchisch aufgebauten Modulen Stückgutumschlaghalle und KV-Terminal. Darin sind Strategien und grundlegende Vorgänge im GVZ hinterlegt. Dynamisch wird das Modell erst durch die Einstellung und Variation der Parameter sowie den Zugriff auf Ressourcen, wobei besonders die Größen Kapazität, Zeit und Kosten im Vordergrund stehen und zur Leistungsbetrachtung im GVZ beitragen.

Schnittstellen existieren sowohl zwischen den einzelnen Hierarchieebenen als auch zu externen, nicht modellimmanenten Quellen. Diese externen Schnittstellen versorgen das Modell mit den notwendigen Informationen über alle Eingangsströme des GVZ.

Die Ströme werden durch einen **Generator** erzeugt, der dem Modell Werte liefert, die entweder aus statistisch erhobenen oder der Praxis entnommenen Daten bestehen. Diese Daten bilden die Basis aller im Modell durchgeführten Operationen und ermöglichen die realitätsnahe Simulation der Prozesse in einem GVZ. Durch den Generator steht an den Modellgrenzen eine frei generierbare Systemlast zur Verfügung, die es ermöglicht, in der experimentellen Phase Ergebnisse zu unterschiedlichen Fragestellungen zu erlangen.

Die **Steuerungsebene** weist den Ressourcen Initialwerte zu und verteilt die vom Generator erzeugten Daten an die Funktionseinheiten.

Ressourcen sind in der Prozesskettentheorie als Bestände, Flächen, Personal, Arbeitsmittel, Arbeitshilfsmittel und Organisationsmittel definiert. In Güterverkehrszentren können, dieser Definition entsprechend, Wechselbrücken, Container, Umschlagpersonal, Dispositionspersonal, Kommunikationseinrichtungen, Züge, Umschlagmittel wie Spreader und Wechselbrückentransportwagen, Verkehrsflächen, Umschlagflächen, Lagerflächen, Verwaltungs- und Sozialflächen sowie Servicebereiche und Gemeinschaftsanlagen zu den Ressourcen gezählt werden. Im ProC/B-Modellierungsparadigma werden Ressourcen durch Server, Counter und Storages modelliert, an denen die Größen Auslastung (nur am Server), Durchsatz, Population, Durchlaufzeit und Zustand (nur am Storage) gemessen werden können.

Die **Leistungsobjekte** des Modells sind Wechselbrücken, Paletten und Kisten. Kisten und Paletten werden im GVZ zur Ladung einer Wechselbrücke oder eines Containers konsolidiert oder aus einer Ladung aufgebrochen und durch die Transport-, Umschlag- und Lagerungsprozesse geschleust. An den Leistungsobjekten können Verweilzeiten an einzelnen Bedienstationen und Durchlaufzeiten durch das gesamte Modell bestimmt werden.

Die aus der Praxis gewonnenen Prozessketten werden als Dienste in **Funktionseinheiten** zusammengefasst. Nachdem sie so in das Modell im ProC/B-Paradigma eingebunden worden sind, können sie über Parameter und Variablen gesteuert auf die Ressourcen oder auf andere Funktionseinheiten, die hierarchisch auf einer anderen Ebene liegen, zugreifen.

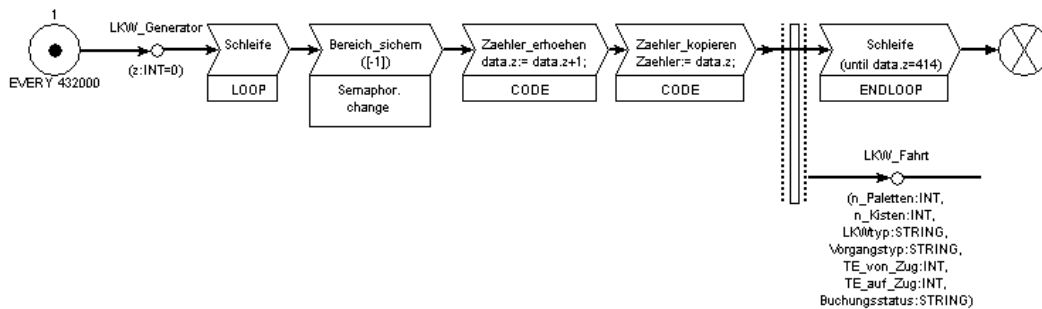


Abb. 3: Prozesskette LKW_Generator.

Die in der folgenden Modellbeschreibung verwendeten Parameter stellen *eine* Variante des GVZ-Modells dar. Diese Parameter können für verschiedene Analysedurchläufe variiert werden, womit unterschiedliche Ausprägungen eines GVZ und seiner Systemlasten betrachtet werden können.

2.4.3 FE GVZ_Generator

Die FE GVZ_Generator beschreibt die Steuerungsebene des Modells. Sie verfügt über drei Prozessketten. Die Prozesskette LKW_Generator (vgl. Abb. 3) erzeugt alle 5 Tage (432000 Sekunden) einen Prozess, der 414 Prozesse auf der Kette LKW_Fahrt initiiert, wovon jeder Prozess je einem LKW entspricht. Die hier verwendeten Werte entstammen aus Datenaufnahmen an dem zu modellierenden System. Die indirekte Erzeugung der LKW_Fahrt-Prozesse ermöglicht es, ihre Zwischenankunftszeiten aus einer Liste zu entnehmen (vgl. [BBF+03]).

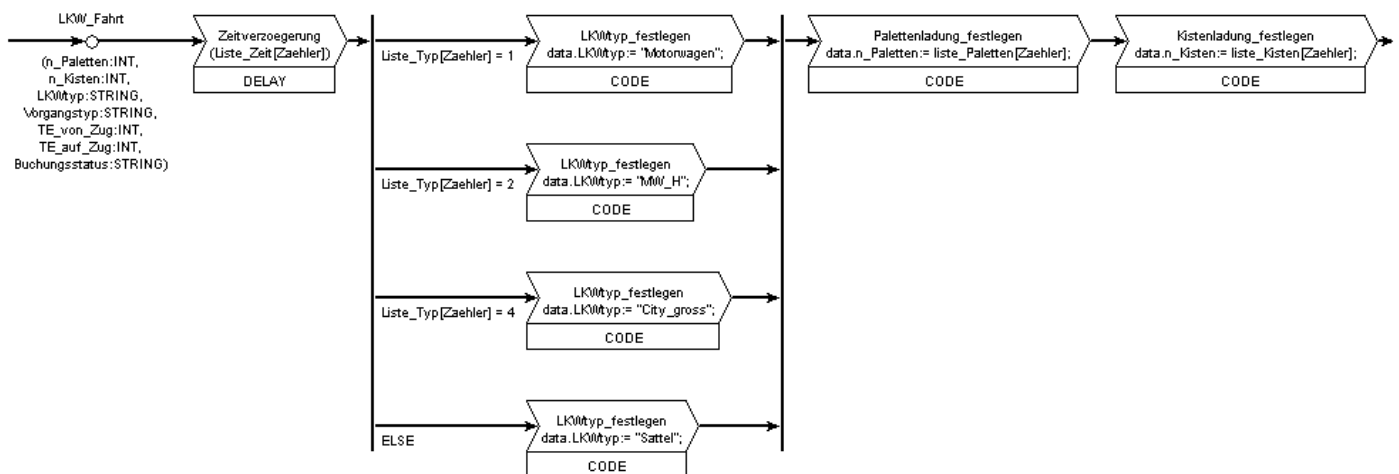


Abb. 4: Einlesen von LKW-Daten aus der Liste (Ausschnitt aus der Prozesskette LKW_Fahrt).

Die Prozesskette LKW_Fahrt dient dazu, Daten zum LKW-Typ, zur Beladung und zum Umschlagstyp (Beladen/Entladen, Stückgut-/KV-Umschlag) festzulegen (vgl. Abb. 4). Diese Daten sind als Liste in das Modell integriert und werden durch die Prozesse der Kette LKW_Fahrt in prozesslokale Variablen eingelesen. Anschließend wird entsprechend der Listendaten in die unterschiedlichen Bereiche des GVZ (vgl. Abb. 5), die verschiedene auszuführende Umschlagstätigkeiten modellieren, verzweigt.

Die Prozesskette Zug_Generator (vgl. Abb. 6) erzeugt ein Mal täglich einen Prozess, der einen Zug repräsentiert. Der Zug wird danach mit Hilfe des Dienstes Zug_Abfertigung der FE KV_Terminal umgeschlagen.

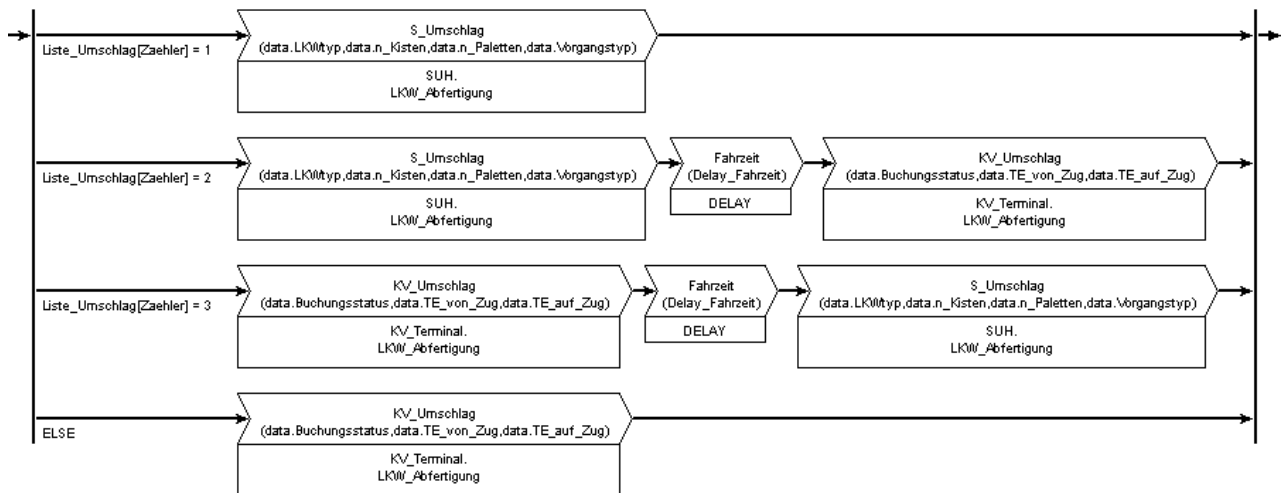


Abb. 5: Verzweigen in die GVZ-Bereiche (Ausschnitt aus der Prozesskette LKW_Fahrt).

2.4.4 Stückgutumschlaghalle (SUH)

2.4.4.1 Allgemeines zur SUH

In dem Modul Stückgutumschlaghalle des GVZ werden palettierte Güter und Kisten, die palettiert werden können, sowie Gitterboxen und ähnliche Transporthilfsmittel von einem LKW auf einen anderen umgeschlagen. Es dient außerdem der Zwischenlagerung und Pufferung, um einen störungsfreien Umschlag zu gewährleisten. Im ablaufenden Umschlagprozess müssen Materialflüsse und Informationsflüsse miteinander verknüpft werden.

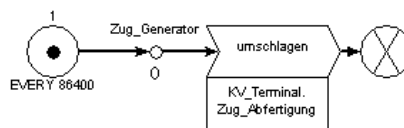


Abb. 6: Prozesskette Zug_Generator.

Zu den Eingangsströmen einer Stückgutumschlaghalle zählen der Nahverkehrseingang, der Fernverkehrseingang und der eingehende KV-Anbindungsverkehr. Die Ausgangsströme sind korrespondierend zu den Eingangsströmen als Nahverkehrsausgang, Fernverkehrsausgang und als abgehender KV-Anbindungsverkehr definiert. Die Güter, die aus diesen Eingangs- und Ausgangsströmen resultieren, werden in der Stückgutumschlaghalle umgeschlagen.

Der Umschlagbereich einer Stückgutumschlaghalle gliedert sich in die 3 Teilbereiche Administrationsbereich/Speditonsleitstand, Ladebereich und Lagerbereich auf.

2.4.4.2 Beschreibung des Modells

Die Stückgutumschlaghalle wurde durch drei FEs modelliert. Die ankommenden Verkehrsträger werden an die FE SUH übergeben, durch die zunächst administrative Tätigkeiten ausgeführt werden. Anschließend erfolgt mit Hilfe der FE SUH_Ladebereich die Vergabe eines Umschlagplatzes und schließlich findet innerhalb der FE SUH_Lagerbereich der Umschlag statt.

2.4.4.2.1 FE SUH

Die FE SUH (vgl. Abb. 7) beschreibt die Abfertigung eines LKW an der Stückgutumschlaghalle. Nachdem ankommende LKW einen Parkplatz im Administrationsbereich reservieren konnten, fahren sie zur Administration und belegen einen Mitarbeiter (durch das Storage Administration modelliert). Die Parkplätze in diesem Bereich sind in der Realität betonierte Flächen, die variabel vergeben werden.

Nachdem sich der Fahrer in der Administration gemeldet und (entsprechend des Vorgangstyps) die Be- oder Entladeliste bekommen hat, verlässt der LKW den Administrationsbereich und somit seinen Parkplatz wieder, fährt weiter in den Ladebereich und wird umgeschlagen.

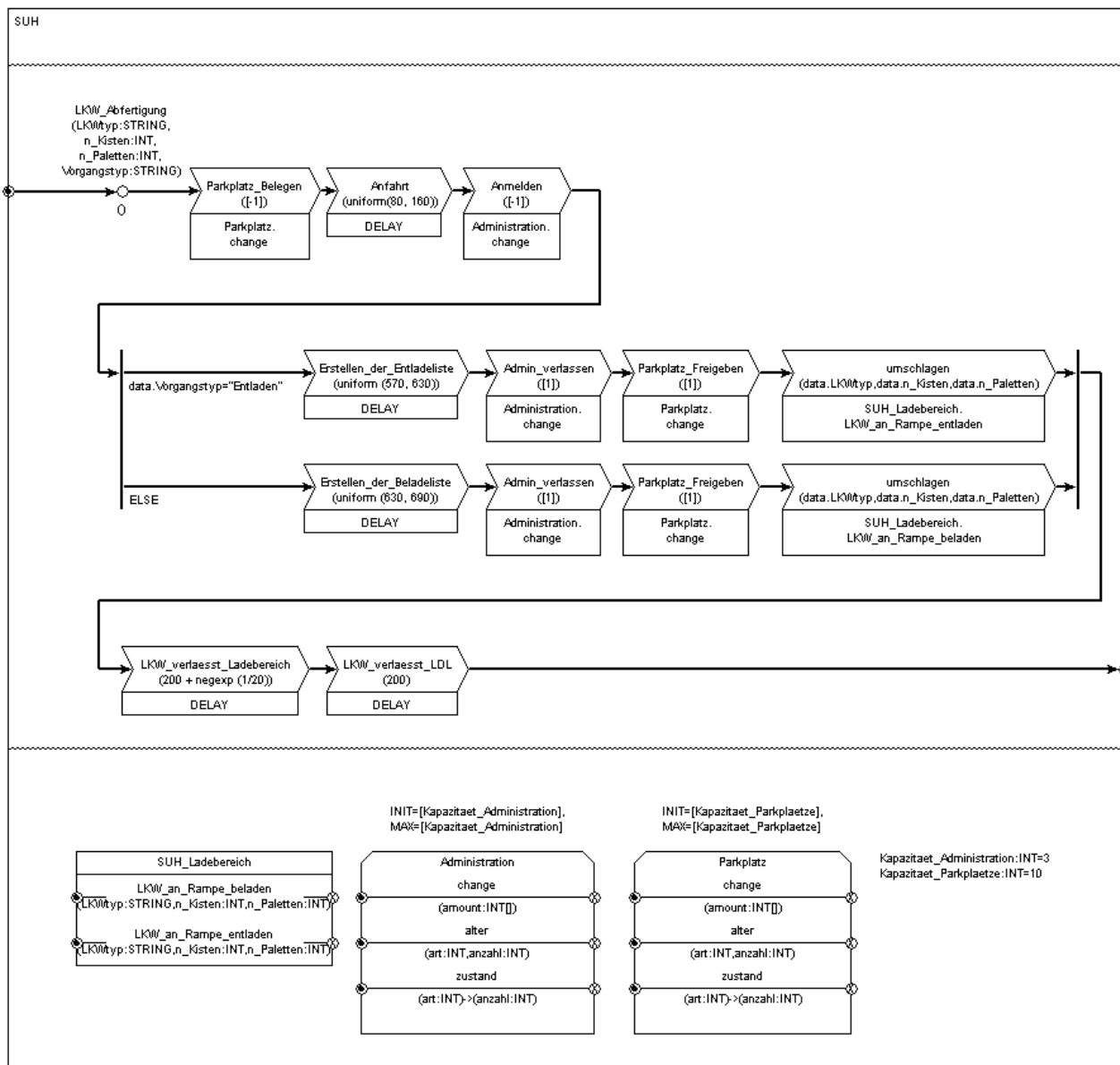


Abb. 7: FE SUH.

2.4.4.2.2 FE SUH_Ladebereich

Der Ladebereich wird durch einen eigenen Funktionsbaustein realisiert. Im Ladebereich dockt das Transportfahrzeug, das ein Nahverkehrs-LKW, ein Gliederzug, ein Solofahrzeug oder ein Sattelschlepper sein kann, an eine freie Rampe an. Hier muss das Modell gewährleisten, dass die unterschiedlichen Transportfahrzeuge auch unterschiedlich behandelt werden und ggf. unterschiedlich viele Rampen belegen. Wenn keine Rampe frei ist, muss das Transportfahrzeug in einem Wartebereich solange warten, bis eine Rampe verfügbar ist. Die entstandenen Wartezeiten sollten möglichst gering sein und sind somit ein Indikator dafür, ob die Rampenanzahl einer Stückgutumschlaghalle hinreichend groß dimensioniert ist. Wenn das Transportfahrzeug an die Rampe angedockt hat, wird es be- oder entladen.

Die FE SUH_Ladebereich verfügt hierzu über zwei Prozessketten, je eine zur Modellierung der Beladung (`LKW_an_Rampe_beladen`) und der Entladung (`LKW_an_Rampe_entladen`). Diese sind bis

auf die Aufrufe von Be- bzw. Entladediensten identisch aufgebaut. In den Ketten besteht jeweils eine Verzweigung hinsichtlich des Fahrzeugtyps, da für Fahrzeuge mit Hänger zwei Rampen und zwei Umschlagsvorgänge nötig sind. Abb. 8 zeigt die Modellierung des Wartebereichs mit der Verzweigung bzgl. der unterschiedlichen Fahrzeugtypen. Die Textausgabe dient zur Erstellung eines Traces über zu lange Wartezeiten.

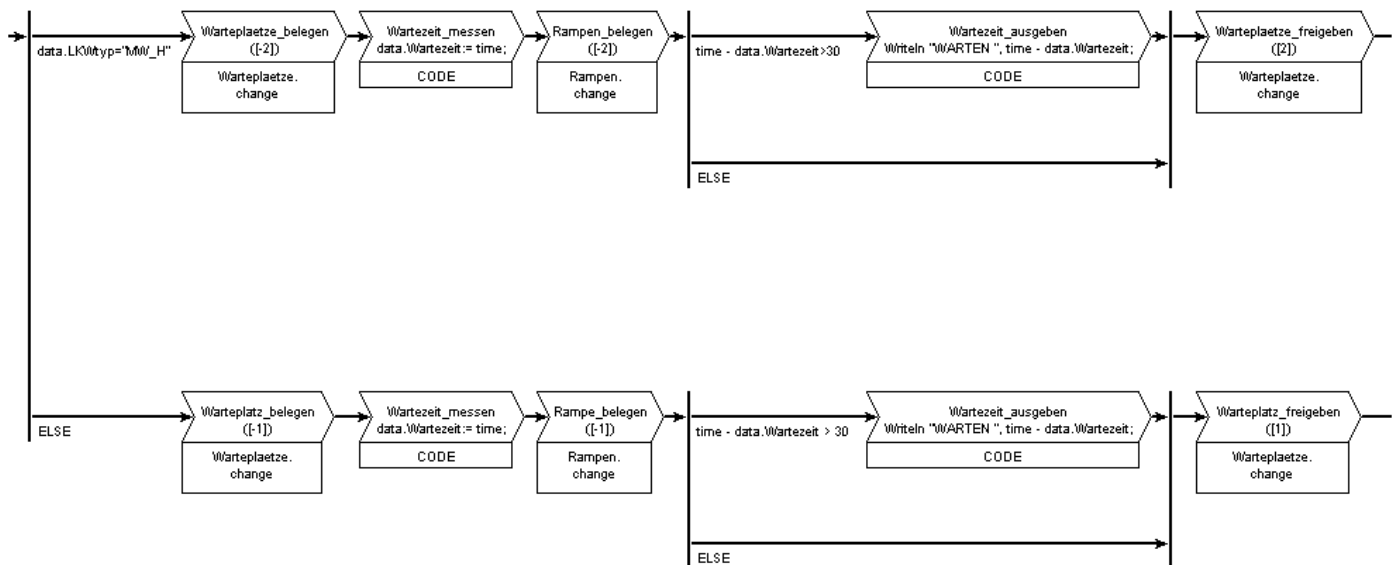


Abb. 8: Warten auf die Rampenzuteilung (Ausschnitt aus der Prozesskette LKW_an_Rampe_beladen).

Abb. 9 zeigt die Modellierung des Rampenbereichs. Bei Fahrzeugen mit Hänger werden im Gegensatz zu den übrigen Fahrzeugen zwei parallele Umschlagvorgänge durchgeführt (je einer für den Motorwagen und den Hänger). Dementsprechend müssen Hängerzüge nach dem Trennen von Motorwagen und Hänger einen zusätzlichen Fahrweg mit dem Motorwagen zurücklegen (PKE Motorwagen_an_Rampe_setzen) und auch anschließend den Hänger wieder aufnehmen (PKE Haenger_von_Rampe_holen). Für den Umschlag werden entsprechende Dienste der FE SUH_Lagerbereich aufgerufen und die dafür notwendigen Informationen übergeben.

2.4.4.2.3 FE SUH_Lagerbereich

Im Lagerbereich entnehmen Ladegruppen die Ladung (Paletten und Kisten) aus dem Transportfahrzeug. Als Entladehilfsmittel stehen im Innenbereich der Rampen Handgabelhubwagen bereit bzw. werden von den Ladegruppen mitgeführt. Bei der Entladung wird die Ladung in den Innenbereich der Rampe gestellt, wo sie einem Relationsplatz zugeordnet wird. Zu diesem wird sie von einem Arbeiter aus der Ladegruppe verbracht. Bei der Beladung der LKW wird die Ladung von den Relationsplätzen in den Innenbereich der Rampe gestellt, wo sie überprüft und daraufhin von dort in das Transportfahrzeug geladen wird.

Die FE SUH_Lagerbereich verwendet 5 Sub-FEs (siehe Abb. 10). Das Storage Ladegruppe modelliert das Ladepersonal, Gabelstapler die beim Umschlagen verwendeten Fahrzeuge und Lager_Paletten das Lager der SUH, in das Güter stets palettiert eingelagert werden. Das Storage Lager_Paletten_Reservierung sowie der Counter Semaphor_Lager_Daten werden benötigt, um einen Abweisungsmechanismus modellieren zu können, der nur LKW zum Umschlag zulässt, die das Lager nicht überlasten, damit Deadlocksituationen in Kombination mit der Ressourcenbelegung vermieden werden (vgl. Abb. 11).

Die FE SUH_Lagerbereich verfügt ebenso wie die FE SUH_Ladebereich (siehe Kapitel 2.4.4.2.2) über je eine Prozesskette zur Modellierung der Be- und Entladung (PK Entladevorgang bzw. Beladevorgang). Diese unterscheiden sich lediglich hinsichtlich der auszuführenden Vorgänge an den Storages (hinzufügen bzw. entnehmen) sowie bei der Palettierung (nur beim Einlagern) oder Depalettierung (nur beim Auslagern) von Kisten.

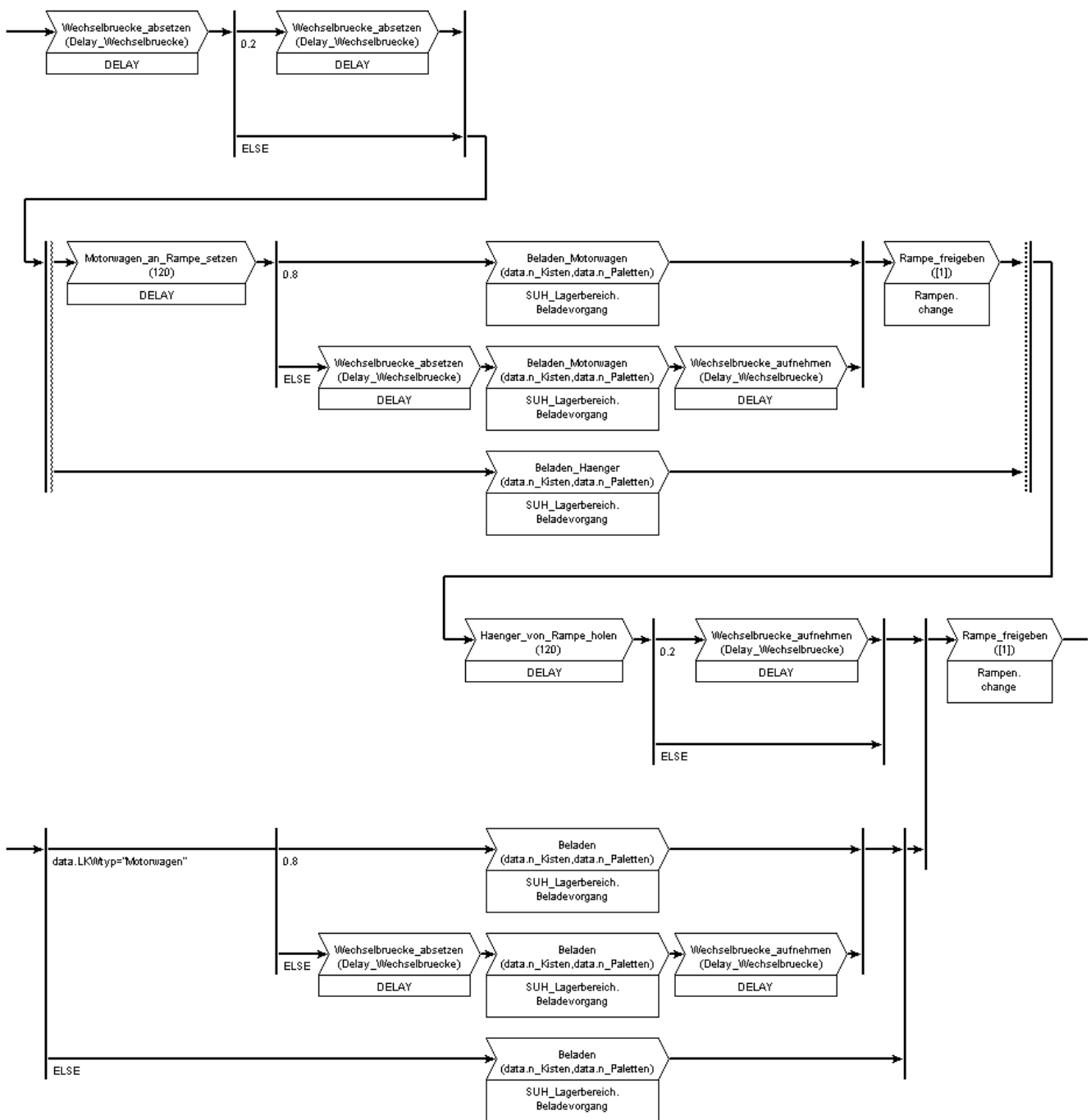


Abb. 9: Beladen im SUH-Ladebereich (Ausschnitt aus der Prozesskette LKW_an_Rampe_beladen; obere Alternative für LKW mit Hänger, untere Alternative für sonstige Fahrzeuge).

Die Ketten starten mit der Überprüfung, ob der Umschlagsvorgang durchgeführt werden kann (vgl. Abb. 11). Dazu wird zunächst der aktuelle Lagerbestand mit Hilfe des Storage Lager_Paletten_Reservierung ermittelt.

Ist beim Einlagern der noch zur Verfügung stehende Raum im Lager ($Kapazitaet_Paletten - data.Lagerbestand$) mindestens so groß wie der für den Umschlag benötigte ($data.n_Paletten$ für palettiert angelieferte Güter und $(data.n_Kisten + 3) // 4$ für in Kisten gelieferte Güter), so werden die umzuschlagenden Mengen auf die lokalen Variablen Palettenzaehler und Kistenzaehler übertragen und der Raum reserviert. Anderenfalls werden die beiden Zähler auf 0 gesetzt, was mit Hilfe des darauffolgenden Oder-Konnektors zum Abweisen des gesamten Umschlagwunsches führt.

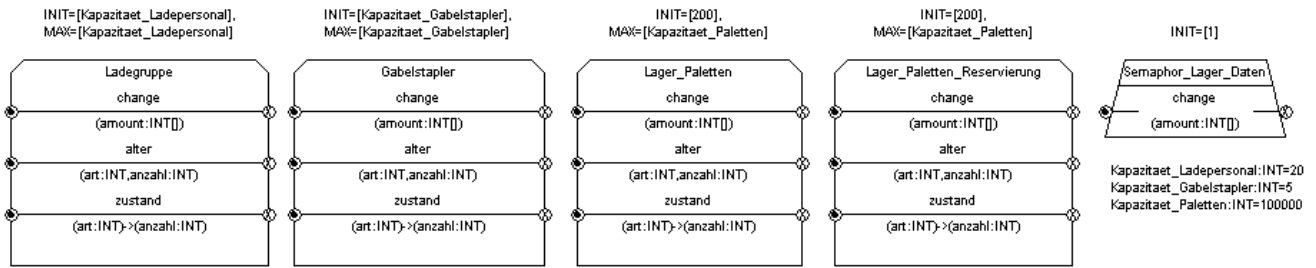


Abb. 10: Sub-FEs der FE SUH_Lagerbereich

Beim Auslagern verläuft die Überprüfung analog, jedoch wird auf die ausreichende Verfügbarkeit von eingelagerten Gütern geprüft.

Für den Umschlag wird je eine Person der Ladegruppe benötigt, die während der gesamten Umschlagzeit eines Fahrzeuges exklusiv genutzt wird (vgl. Abb. 12). Der Umschlag von Kisten und Paletten findet parallel in je einer Schleife statt. Jeder Durchlauf der Palettenschleife schlägt eine Palette um, jeder Durchlauf der Kistenschleife je 4 Kisten bzw. die Restmenge, wenn diese kleiner als 4 ist (siehe Abb. 12).

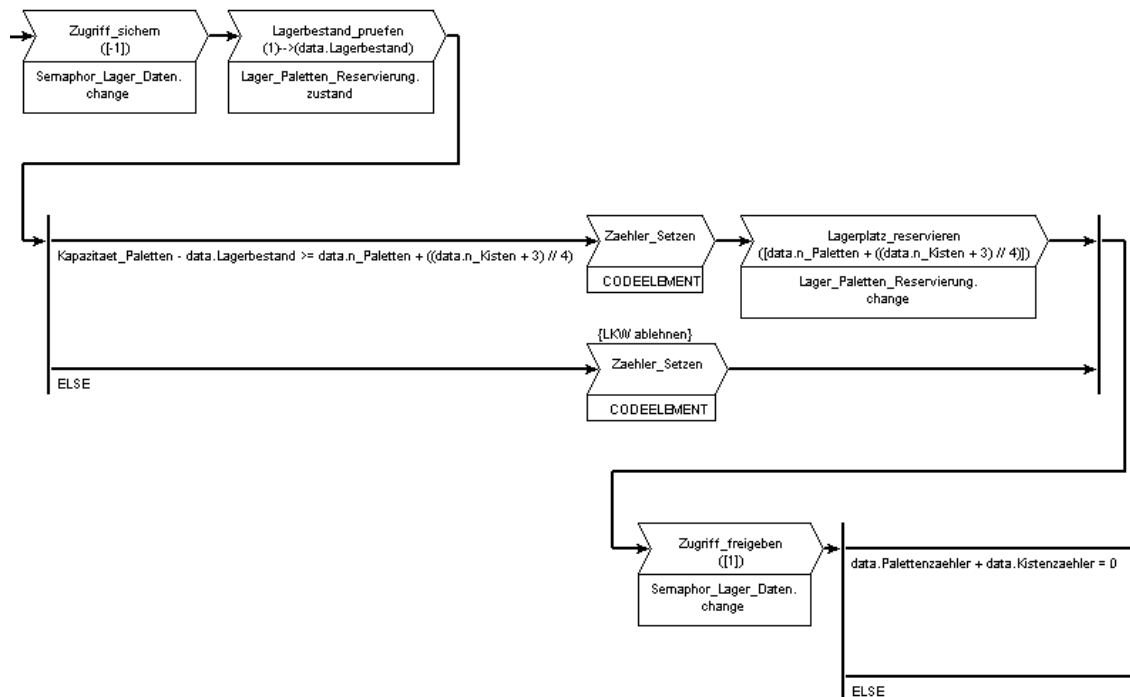


Abb. 11: Überprüfung auf Durchführbarkeit des Umschlags (Ausschnitt aus der Prozesskette Entladevorgang).

Der jeweilige Umschlagvorgang der Schleifendurchläufe ist in Abb. 13 (für Paletten) und Abb. 14 (für Kisten) dargestellt. Da die Kisten nur auf Paletten umgeschlagen werden, sind diese beim Einlagern zu je bis zu 4 Stück zu palettieren (PKE Kisten_palettieren in Abb. 14). Beim Auslagern werden die Kisten nach der Entnahme aus dem Lager dementsprechend depalettiert. Anschließend werden die Kisten den Paletten entsprechend umgeschlagen. Zusätzlich zum Personal wird bei 2,5% aller umzuschlagenden Paletten ein Gabelstapler benötigt, der für jeweils einen Schleifendurchlauf angefordert wird (vgl. Abb. 13).

Um einen Lagerbestandsgraphen erstellen zu können, werden in jedem Schleifendurchlauf Daten zum Lagerbestand ausgegeben.

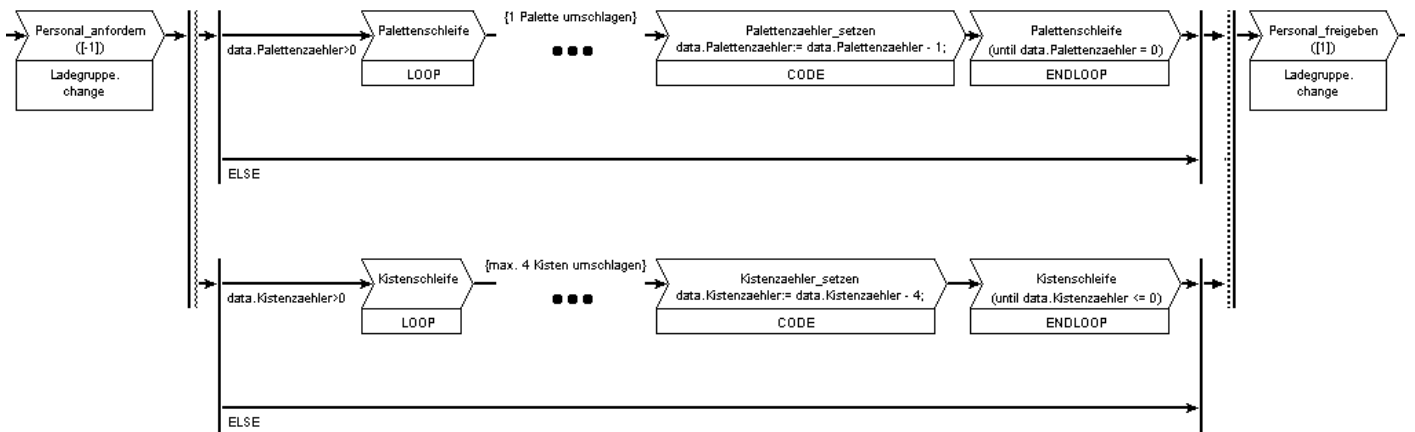


Abb. 12: Grundstruktur des Umschlags von Paletten und Kisten (Ausschnitt aus der Prozesskette Entladevorgang).

2.4.5 KV-Terminal

2.4.5.1 Allgemeines zum KV-Terminal

In dem Modul KV-Terminal werden als Kernfunktion KV-spezifische Güter von der Schiene auf die Straße und umgekehrt umgeschlagen. KV-spezifische Güter können in diesem Zusammenhang Wechselbrücken und alle Arten von Containern sein. Der Umschlag im KV-Terminal geschieht mittels eines oder mehrerer Portalkrane. Weitere Elemente des KV-Terminals sind das Gate, Parkflächen, Verkehrsflächen, Ladespuren, Gleisanlagen, Abstellspuren und Lagerflächen. Im KV-Terminal treffen die beiden Verkehrsträger LKW und Zug aufeinander und schlagen untereinander Güter um. Diese Prozesse werden von der Administration im Gate disponiert, sodass die beiden Verkehrsträger den Umschlag synchron abwickeln können. Des Weiteren übernimmt die Administration die Disposition der Lager- und Abstellflächen. Somit stellt die Administration ein zentrales Element im KV-Terminal dar und ist auf die nötige Information zu allen diesen Prozessen angewiesen. In der Realität muss diese Information bereitgestellt, übermittelt und in der Administration mit Hilfe geeigneter Kommunikationssystemen verknüpft und mittels Datenverarbeitungssystemen ausgewertet und verarbeitet werden, sodass aus diesen Daten Aktionen im Terminal generiert werden können. Die eingesetzten Technologien sind Datennetze, Funkübertragung, Telefon, Videosysteme, GPS, Datenbanksysteme und Simulationen. Im Modell sind diese informationstechnischen Verknüpfungen durch die Schnittstellen, Verweise und Verknüpfungen der Prozessketten und ihrer Elemente hinterlegt.

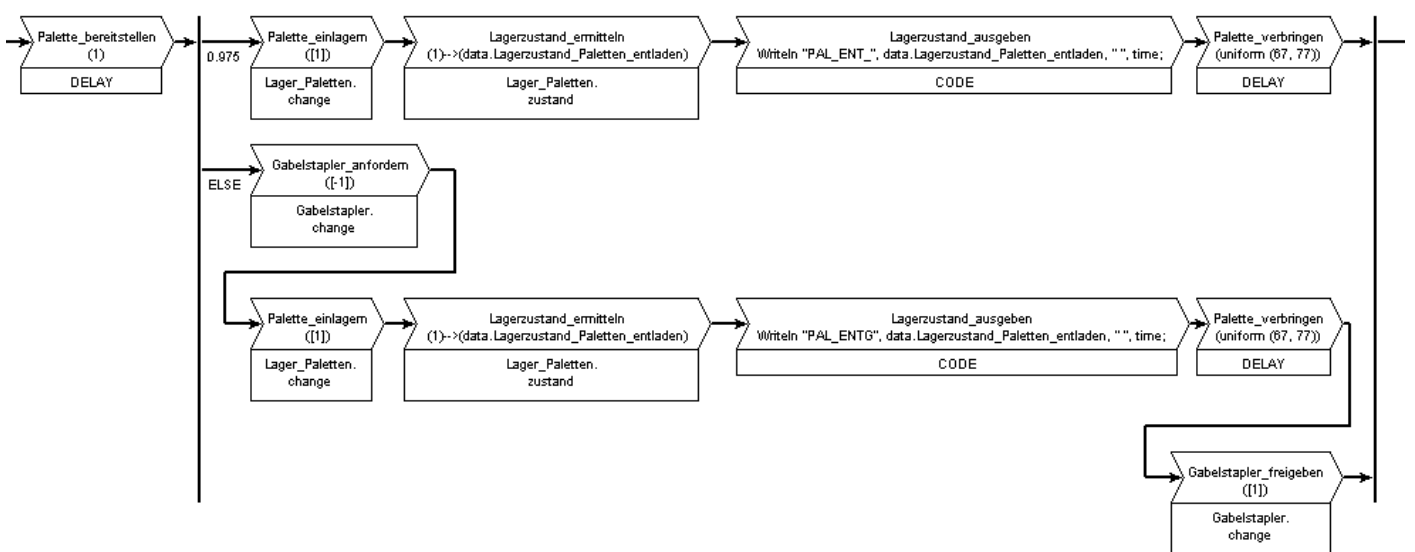


Abb. 13: Umschlag einer Palette beim Entladen (Ausschnitt aus der Prozesskette Entladevorgang).

Die im KV-Terminal ankommenden Verkehrsströme setzen sich aus Zügen und LKW zusammen. Die LKW stammen hauptsächlich aus dem KV-Anbindungsverkehr, können aber auch ohne Abfertigung über die Stückgutumschlaghalle direkt am KV-Terminal ankommen und schon komplett vorsortierte Wechselbrücken oder Container anliefern. Die Züge liefern Güter von einem GVZ zu einem anderen GVZ, einem anderen Containerterminal oder direkt von einem Unternehmen und bringen sie auch wieder zu diesen Quellen. Im Modell werden die Verkehrsträger über einen Generator erzeugt und an die nachstehenden Ebenen im Modell übergeben.

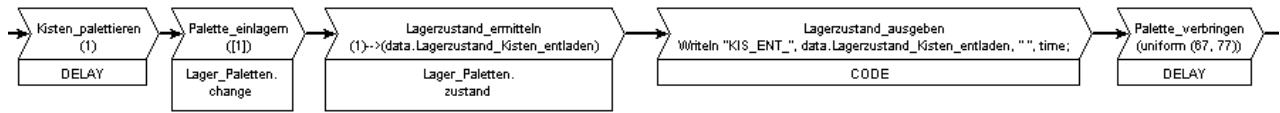


Abb. 14: Umschlag von bis zu 4 Kisten beim Entladen (Ausschnitt aus der Prozesskette Entladevorgang).

2.4.5.2 Beschreibung des Modells

Die Kernbausteine des KV-Terminals sind die Administration und der Umschlag. Aus diesem Grund wurden die beiden Elemente als eigenständige Ebenen im Modell realisiert. Die im Generator erzeugten Verkehrsträger werden an die übergeordnete FE `KV_Terminal` übergeben, in welcher die Wartezeiten und Ressourcenverwaltung bei Anmeldung am Gate hinterlegt sind. Von hier aus werden die Daten zunächst an die FE `Identifikation` übergeben, in der die Datenverarbeitung mit den dazugehörigen Ressourcen und Prozesszeiten stattfindet. Die Parameter und Strukturen sind sowohl auf Terminals mit hoher Technologiedurchdringung als auch für niedrig technologisierte Terminals ausgelegt. Die Zeiten für die Bearbeitung der Umschlagaufträge sind in einem statistischen Rahmen festgelegt. Nach der Freigabe für den Umschlag werden die Verkehrsträger aus der FE `KV_Terminal` heraus an die Sub-FE `KV_Umschlag` übergeben, in der der eigentliche Umschlag von LKW auf Zug oder Zug auf LKW stattfindet.

2.4.5.2.1 FE `KV_Terminal`

Die FE `KV_Terminal` (siehe Abb. 15) verwendet neben den Sub-FEs `Identifikation` (siehe Kapitel 2.4.5.2.2) und `KV_Umschlag` (siehe Kapitel 2.4.5.2.3) die durch Storages modellierten Einheiten `Umschlagplaetze` und `Gleise` zur Verwaltung des für die Fahrzeuge im KV-Terminal benötigten Raumes. Ferner ist ein durch ein Counter-elemente ein Semaphore definiert, der bei der Abweisung von Zügen eingesetzt wird.

In der FE `KV_Terminal` sind drei Prozessketten definiert. Die PK `LKW_Abfertigung` modelliert das Umschlagen eines LKW. Dazu wird nach der Identifikation des LKW ein Umschlagplatz belegt, dann über den Dienst `umschlagen_LKW` der FE `KV_Umschlag` der LKW umgeschlagen, der Umschlagplatz freigegeben und das `KV_Terminal` verlassen. Für die Züge ist innerhalb der PK `Zug_Abfertigung` ein Abweisungsmechanismus modelliert, der verhindern soll, dass sich am KV-Terminal Züge stauen, die auf ein freies Gleis warten. Für den Fall, dass ein Zug in das KV-Terminal fahren darf, wird analog zu den LKW, ein Gleis belegt, dann über den Dienst `umschlagen_Zug` der FE `KV_Umschlag` der Zug umgeschlagen, das Gleis freigegeben und das `KV_Terminal` verlassen. Die PK `Messung_Umschlagplaetze` dient ausschließlich der Ergebnisdarstellung (Trace der Anzahl freier Umschlagplätze).

2.4.5.2.2 FE `Identifikation`

Die FE `Identifikation` (siehe Abb. 16) modelliert mit dem Dienst `LKW_Identifikation` die Anmeldung der LKW am KV-Terminal. Unterschieden wird dabei zwischen LKW, die einen Umschlag vorgebucht haben und jenen, die ohne Anmeldung das KV-Terminal anfahren (definiert durch die prozesslokale Variable `Buchungsstatus`). Während die Abwicklung bei den vorangemeldeten LKW lediglich wenige kurze Verzögerungen umfasst, werden bei den nicht angemeldeten LKW die über

Counter definierten Ressourcen Gateparkplatz und Gatepersonal verwendet. Dabei wird zunächst ein Parkplatz belegt und ein Mitarbeiter angefordert, die Frachtpapiere werden abgegeben, der genaue Ort des Umschlags wird gesucht, und anschließend werden sowohl der Mitarbeiter als auch der Parkplatz wieder freigegeben.

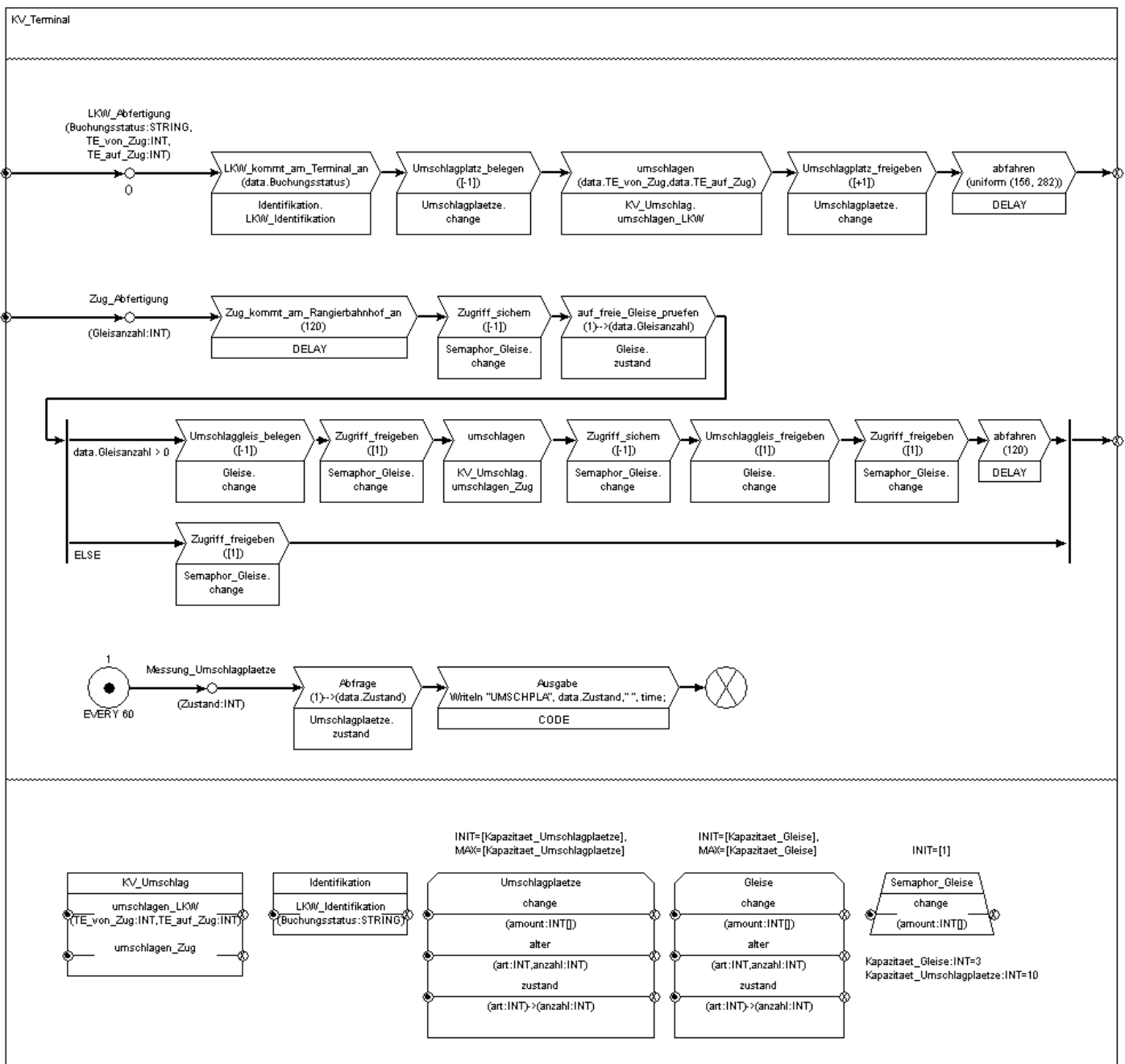


Abb. 15: FE KV_Terminal.

2.4.5.2.3 FE KV_Umschlag

In der FE `KV_Umschlag` ist das Umschlagen der Transporteinheiten von LKW auf Züge oder Züge auf LKW modelliert. Implizit hinterlegt sind Strategien zur Reihenfolge der Abfertigung und zur Bevorzugung des Direktumschlags. Direktumschlag bedeutet, dass bei ankommenden Verkehrsträgern zunächst geprüft wird, ob die Transporteinheiten direkt mit einem Kranungsauftrag, ohne sie abzustellen, auf den anderen Verkehrsträger verladen werden können.

Falls die Verkehrsträger keinen Partner zum Vollzug eines Direktumschlags vorfinden, wird die Ladung auf die Abstellspur abgeladen oder von ihr aufgenommen. Die Umschlagprozesse finden (abgesehen von Abbruchsituationen) so lange statt, bis die durch den Generator vorgegebene Anzahl an Transporteinheiten umgeschlagen oder eingelagert wurde. Daraufhin verlassen die Verkehrsträger die Funktionseinheit KV_Umschlag.

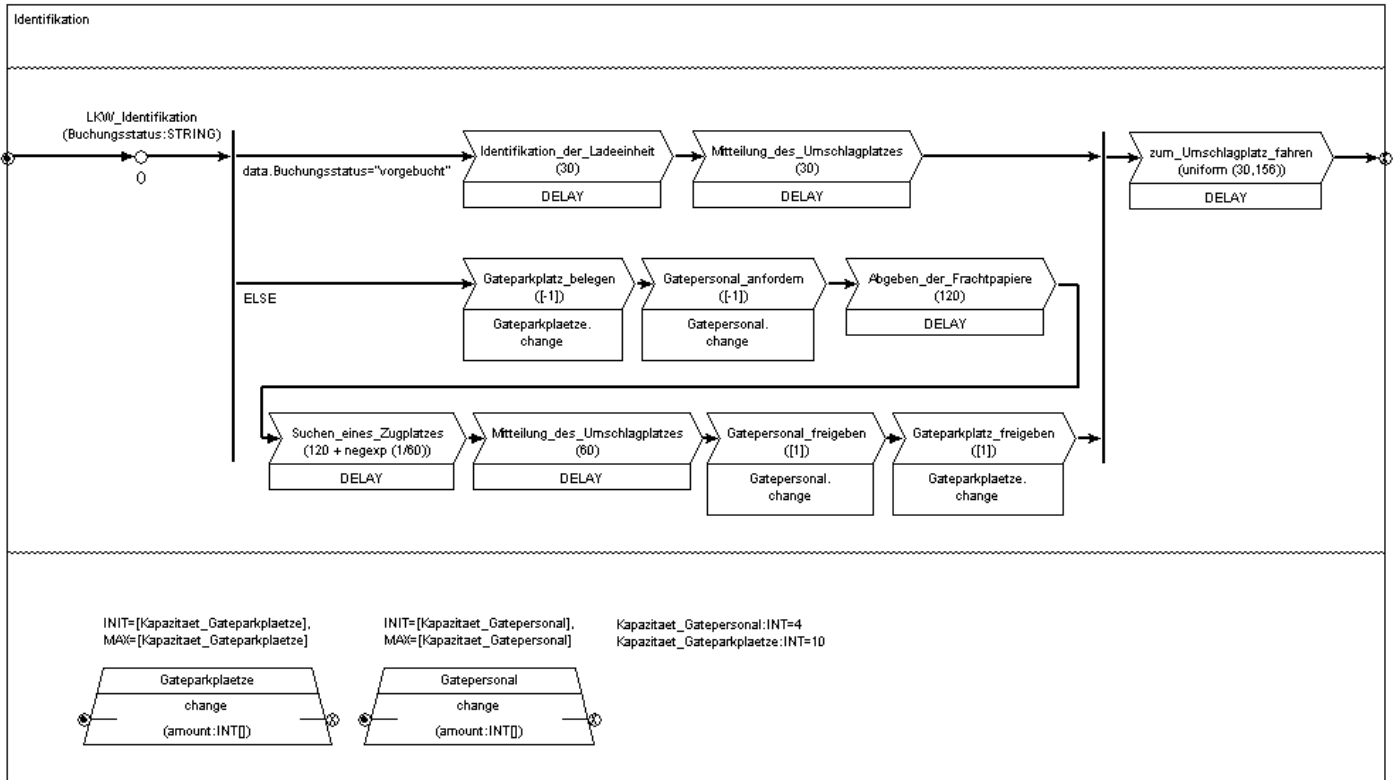


Abb. 16: FE Identifikation.

12 Sub-FEs (vgl. Abb. 17) werden in der FE KV_Umschlag verwendet. Dazu gehören Stages für die im KV-Umschlag verfügbaren Kräne (FE Kraene), für die auf den Abstellspuren zwischengelagerten Transporteinheiten (FE Lager_LKW für auf LKW zu ladende Transporteinheiten sowie FE Lager_Zug für auf Züge zu ladende Transporteinheiten), für die Verwaltung von Ladung während des Direktumschlags (FEs Direktumschlag_auf_LKW bzw. Direktumschlag_auf_Zug), für die Reservierung von Ladung bzw. zugehörigen Fahrzeugplätzen (FEs Ladung_LKW_Reservierung bzw. Ladung_Zug_Reservierung als Summenbildung der zwischengelagerten und direkt umzuschlagenden Ladung) sowie für die Verwaltung der für Transporteinheiten frei zur Verfügung stehenden Fahrzeugplätze (FEs LKW_Plaetze bzw. Zug_Plaetze). Ferner ist neben Semaphoren ein Counter Freiraum_Kontrolle definiert, durch den das Beladen von Zügen nur erlaubt wird, wenn freier Raum auf dem Zug verfügbar ist. Dieser Counter ist nur für die Züge notwendig, da Züge gleichzeitig be- und entladen werden können (was eine derartige Synchronisation erfordert), während LKW zunächst entladen werden, bevor sie beladen werden können.

Die FE KV_Umschlag verfügt neben von zur reinen Traceerstellung verwendeten Prozessketten (vgl. PK Messung_Umschlagplaetze im KV_Terminal, Abb. 15) über zwei die Aktionen innerhalb des KV-Umschlags modellierende Prozessketten. Eine modelliert das Umschlagen in Bezug auf die Züge (PK umschlagen_Zug, die andere in Bezug auf die LKW (PK umschlagen_LKW).

Wie in Abb. 18 dargestellt, verläuft das Umschlagen der Züge bzgl. Auf- und Abladen parallel (die Kette umschlagen_Zug wird mit PK-Konnektoren aufgespalten in die Bereiche Aufladen und Abladen), während im Gegensatz dazu das Umschlagen der LKW sequentiell abläuft, wobei zunächst entladen und dann beladen wird. Ein weiterer wesentlicher Unterschied ist, dass die LKW nicht auf Züge oder freien

Raum im Zwischenlager (= Abstellspur) warten, sondern dann den jeweiligen Vorgang abrechnen. Dadurch wird gewährleistet, dass die FE nicht durch wartende LKW blockiert wird, sondern lebendig bleibt. Im Bereich der Züge wird diese Lebendigkeit durch das Parallelisieren von Be- und Entladevorgänge erreicht. Somit ist garantiert, dass keine untätigen LKW-Prozesse in der FE KV_Umschlag warten und die Zug-Prozesse auf jeden von den LKW-Prozessen durchführbaren Vorgang ordnungsgemäß reagieren können. Ein sich andeutendes Warten der LKW auf die Züge führt zu einem Abbruch des Vorgangs des LKW-Prozesses, ein Warten eines Zuges auf die LKW führt dazu, dass alle durch den Zug momentan nicht bedienbaren Vorgänge (z. B. Aufladen auf einen aktuell vollen Zug) bereits zuvor von den LKW beendet werden, somit neue LKW-Prozesse in den KV-Umschlag nachrücken können und bei stochastischen Auf- bzw. Ablademengen der LKW auch die Auf- und Abladewünsche der Züge erfüllt werden können. Dadurch ergibt sich, dass auch alle Züge die FE KV_Umschlag wieder verlassen können

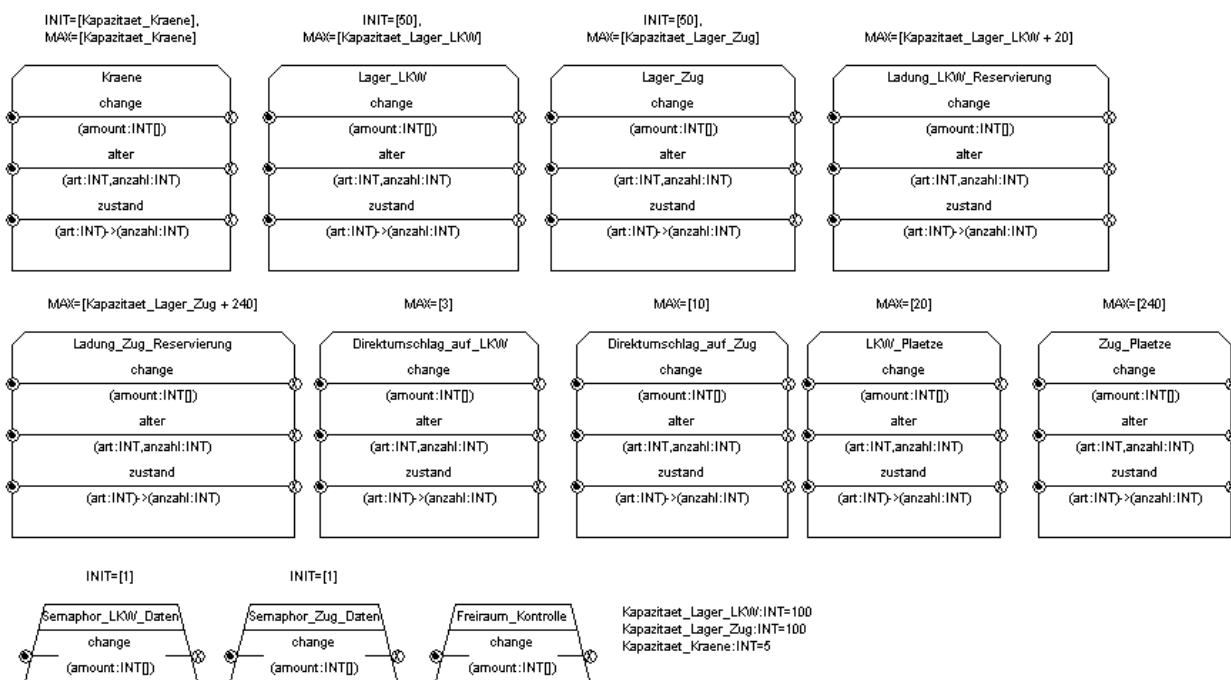


Abb. 17: Sub-FEs der FE KV_Umschlag.

PK umschlagen_Zug

Auf der Prozesskette umschlagen_Zug entspricht jeder Prozess je einem Zug, der 80 Transporteinheiten transportieren kann (jedoch nicht muss, da auch teil- und unbeladene Züge zulässig sind).

Zunächst wird der bei der Ankunft am KV-Terminal für aufzuladende Transporteinheiten auf dem Zug zur Verfügung stehende Platz ermittelt (vgl. Abb. 19). Passen die aufzuladenden Einheiten auf den Zug ohne etwas abzuladen, so entspricht dieser Platz der Anzahl aufzuladender Einheiten (eine darüber hinaus gehende Erhöhung dieses Wertes würde zu einer Überladung des Zuges hinsichtlich der für ihn festgelegten Auflademenge führen). Müssen zunächst Einheiten abgeladen werden, bevor alle aufzuladenden Güter auf den Zug passen, so entspricht der ermittelte Wert dem gesamten aktuell freien Platz auf dem Zug (data.TE_auf_LKW entspricht der mitgebrachten Anzahl und 80 ist die Kapazität eines Zuges, somit ergibt $80 - \text{data.TE_auf_LKW}$ den aktuell freien Platz). Zur Benachrichtigung abladewilliger LKW-Prozesse werden die Zug_Plaetze um den so ermittelten Freiraum erhöht.

Anschließend erfolgt eine Aufspaltung der Kette, so dass nebenläufig ein Zug sowohl be- als auch entladen werden kann (vgl. Abb. 18).

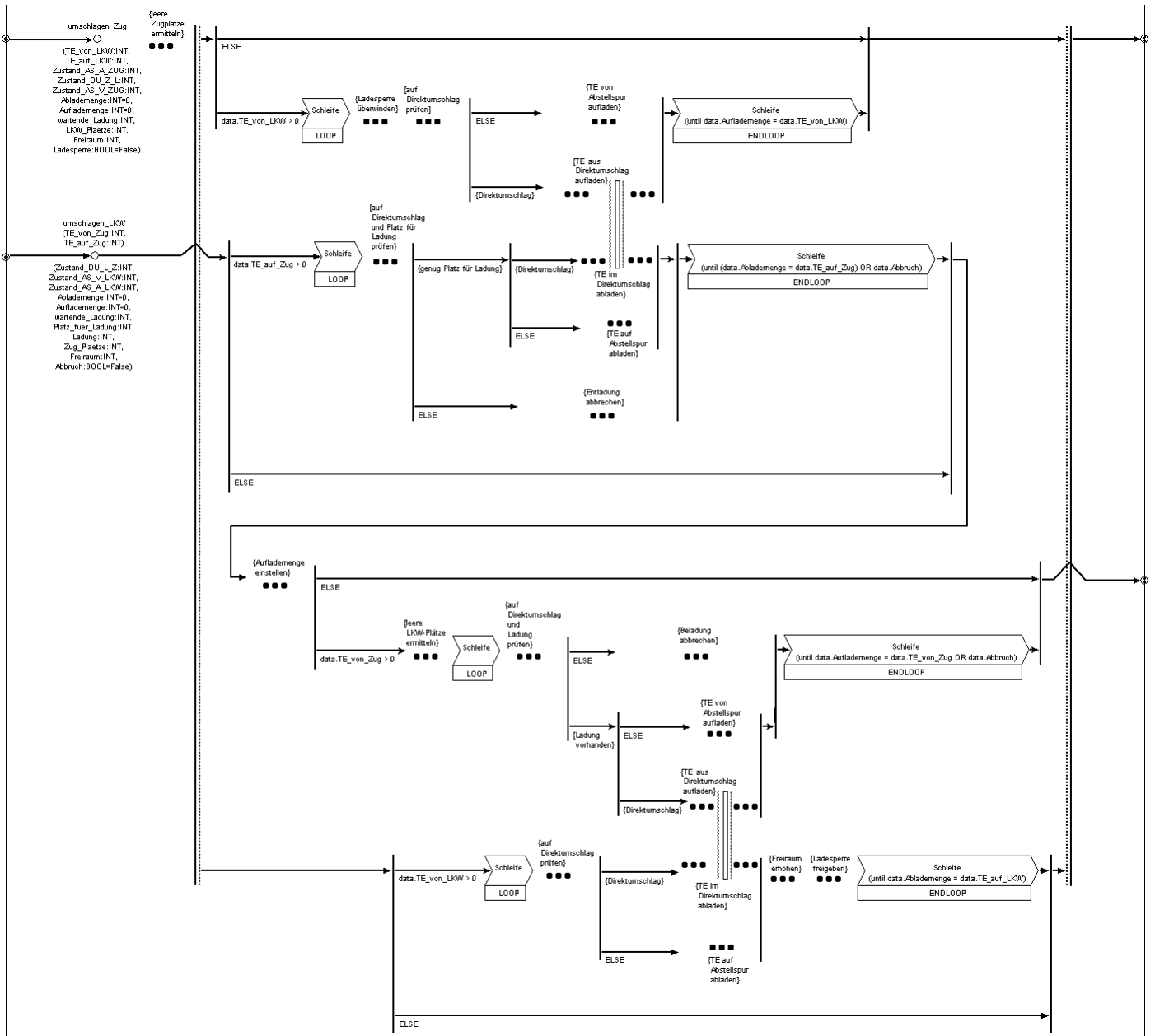


Abb. 18: Modellierungsprinzip der Prozessketten umschlagen_Zug und umschlagen_LKW.

Modellierung des Aufladevorgangs

Das Aufladen erfolgt innerhalb einer Schleife (vgl. Abb. 18), deren einzelne Durchläufe jeweils für die Beladung mit je einer Transporteinheit zuständig sind.

Ist auf den Zug etwas aufzuladen, so wird zunächst der Freiraum auf dem Zug überprüft (vgl. Abb. 20). Ist nicht genügend Freiraum vorhanden, so wird eine Warteschleife durchlaufen, die eine prozess-lokale (= für jeden Zug eigenständige) Variable *Ladesperre* setzt und ein Element der *Freiraum_Kontrolle* entnehmen möchte. Die *Freiraum_Kontrolle*, die nur vom Entladungsteil der PK *umschlagen_Zug* erhöht werden kann, ist eine Einheit, die nur gemeinsam für alle im *KV_Umschlag* befindlichen Prozesse bzw. Züge definiert werden kann. Damit jedoch jeweils der richtige Zug freigegeben wird (= der Entladungsbereich darf ausschließlich den eigenen Zug freigegeben), wird nach dem möglichen Entnehmen aus der *Freiraum_Kontrolle* zusätzlich die lokale Variable *Ladesperre* getestet (denn auch diese setzt der Entladungsbereich bei zusätzlicher Freiraumgewinnung neu). Ist in dieser keine Sperre verzeichnet (*data.Ladesperre* hat dann den Wert *False*), so kann die Warteschleife verlassen werden. Im anderen Fall wird das entnommene Element der *Freiraum_Kontrolle* zurückgegeben, da

es offenbar von einem anderen Zug stammt. Durch die Bearbeitungsreihenfolge des `change`-Aufrufs in der `Freiraum_Kontrolle` ist gewährleistet, dass alle wartenden Zug-Prozesse nacheinander das Element der `Freiraum_Kontrolle` entnehmen können, bis der richtige Zug-Prozess (der einem Zug mit neuem Freiraum entspricht) an der Reihe ist.

Eine Modellierung mit einem allen Zug-Prozessen gemeinsamen Counter musste für das Sperren hinsichtlich einer Beladung erfolgen, da bei fehlendem Freiraum eine Warteschlange für die Zug-Prozesse in das Modell eingefügt werden musste, die durch Counter angeboten wird. Eine Freigabe eines anderen Zug-Prozesses als des eigenen kann lediglich dann erfolgen, wenn auch für den anderen Prozess eine Freigabe erfolgen darf. Dies garantiert jedoch, dass ein weiteres Element zur `Freiraum_Kontrolle` hinzugefügt worden ist, so dass der andere Prozess (oder noch weitere gleichzeitig freigebende) ohne Verzögerung hinsichtlich der Modellzeit auch den eigenen Prozess freigibt.

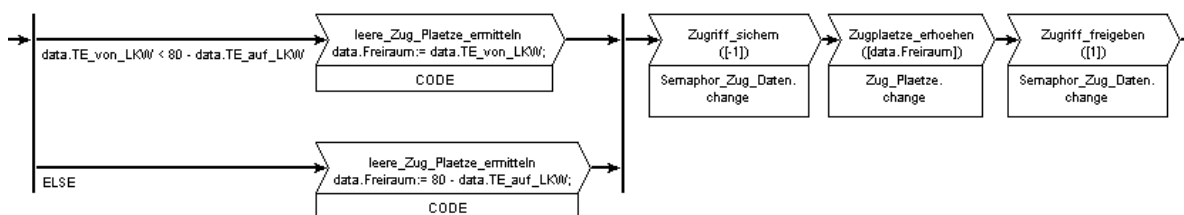


Abb. 19: Ermittlung des für aufzuladende Transporteinheiten auf dem Zug zur Verfügung stehenden Platzes (Ausschnitt aus der Prozesskette `umschlagen_Zug`).

Nachdem feststeht, dass Freiraum vorhanden ist, wird die `Auflademenge` erhöht, Ladung reserviert, der `Freiraum` verringert und auf `Direktumschlag` wartende Ladung geprüft (siehe Abb. 21). Hier zeigt sich die Notwendigkeit für den `Ladung_Zug_Reservierung`-Baustein (die Benutzung des `Ladung_LKW_Reservierung`-Bausteins ist auf die gleiche Weise motiviert), da im Fall, dass aktuell keine Ladung für den Zug zur Verfügung steht, die Warteschlange für die auf Ladung wartende Züge vor der Entscheidung für oder gegen den `Direktumschlag` platziert sein muss. Ansonsten können Fälle eintreten, bei denen Zug-Prozesse auf Ladung von der Abstellspur warten, später ankommende LKW jedoch nur den `Direktumschlag` durchführen möchten. Bei der Auswertung, ob ein `Direktumschlag` durchgeführt wird oder nicht, steht somit bereits fest, dass auf jeden Fall eine Beladung ohne weitere Wartezeit möglich ist.

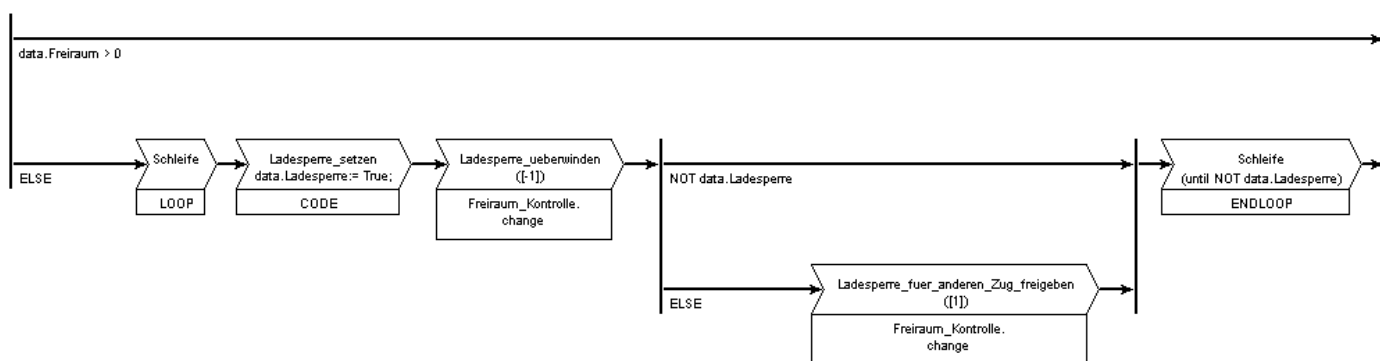


Abb. 20: Freiraumprüfung und Warten an der Ladesperre (Ausschnitt aus der Prozesskette `umschlagen_Zug`).

Beim Aufladen von der Abstellspur, wird die Anzahl der `Zug_Plaetze` um 1 verringert (da dieser durch die folgende Beladung belegt wird), ein Kran angefordert, die Transporteinheit aufgeladen und anschließend der Kran wieder zurückgegeben (vgl. Abb. 21).

Bei Durchführung eines `Direktumschlags` (siehe Abb. 21) wird die Anzahl der `Zug_Plaetze` nicht durch den Zug-Prozess verringert (dafür ist der zugehörige LKW-Prozess zuständig). Jedoch wird eine direkt

umzuschlagende Transporteinheit reserviert und dadurch der Direktumschlag eingeleitet (PKE `Direktumschlag_einleiten`). Ist ein entladender LKW-Prozess gemeinsam mit dem Zug-Prozess zum Direktumschlag bereit (durch den ersten PK-Konnektor in Abb. 21 modelliert), so fordert der LKW-Prozess einen Kran an und lädt die Transporteinheit ab. Anschließend (durch den zweiten PK-Konnektor in Abb. 21 modelliert) wird durch den Zug-Prozess die Transporteinheit auf den Zug aufgeladen und der Kran zurückgegeben. Eine Blockierung des Krans beim Aufladen durch einen Zug-Prozess ist nicht möglich, da stets garantiert ist, dass eine Ladung vorhanden ist, und Kräne nur während des Be- bzw. Entladens durch Zug- bzw. LKW-Prozesse gehalten werden. Hierdurch werden Deadlocks vermieden.

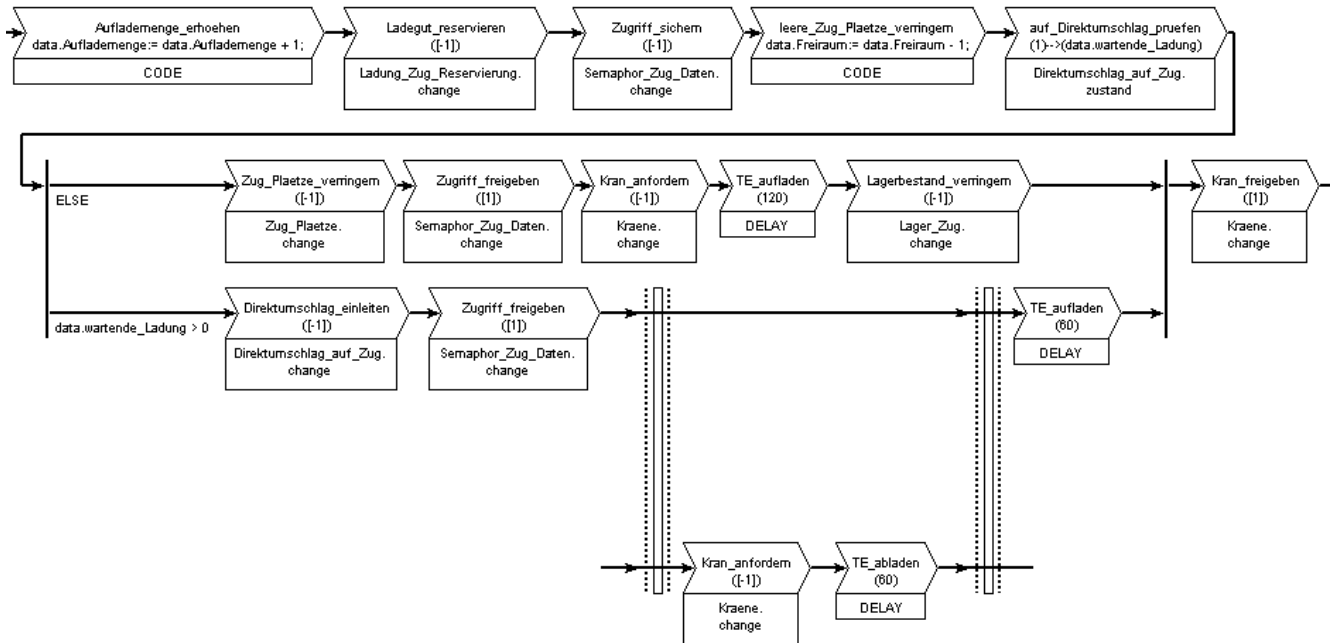


Abb. 21: Beladen eines Zuges mit einer TE (Ausschnitt aus den Prozessketten `umschlagen_Zug` (oben) und `umschlagen_LKW` (unten)).

Nach erfolgter Beladung des Zuges wird wieder an den Anfang der Aufladeschleife gesprungen, falls noch etwas aufzuladen ist. Im anderen Fall ist die Beladung abgeschlossen und, nachdem auch die Entladung abgeschlossen ist, die Ausführung des Dienstes `umschlagen_Zug` beendet.

Modellierung des Abladevorgangs

Das Abladen wird wie das Aufladen ebenfalls innerhalb einer Schleife modelliert (vgl. Abb. 18), wobei jeder Durchlauf dem Abladen einer Transporteinheit dient.

Innerhalb der Schleife wird zunächst Platz für die Ladung auf einem LKW bzw. auf der Abstellspur reserviert (PKE `Platz_für_Ladung_reservieren`) und auf die Möglichkeit eines Direktumschlags geprüft (vgl. Abb. 22).

Kann kein Direktumschlag ausgeführt werden, so wird ein Kran angefordert, die Transporteinheit abgeladen und als für die Aufladung auf einen LKW zwischengelagert registriert (PKE `Lagerbestand_erhoehen`) sowie der Kran freigegeben (siehe Abb. 22).

Im Fall des Direktumschlags (vgl. Abb. 22) wird ein Platz auf einem LKW belegt (die Belegungsmöglichkeit ist durch die vorhergehende Prüfung garantiert) und die Transporteinheit zu denjenigen addiert, die für Direktumschläge auf LKW warten (PKE `fuer_Direktumschlag_bereithalten`). Nachdem ein LKW zusammen mit dem Zug den Direktumschlag beginnen kann (die Gleichzeitigkeit ist durch den ersten PK-Konnektor in Abb. 22 modelliert), fordert der Zug einen Kran an und lädt die Transporteinheit ab. Danach wird die Kontrolle über den Direktumschlag an den LKW abgegeben (die dafür notwendige Synchronisation ist durch den zweiten PK-Konnektor in Abb. 22 modelliert).

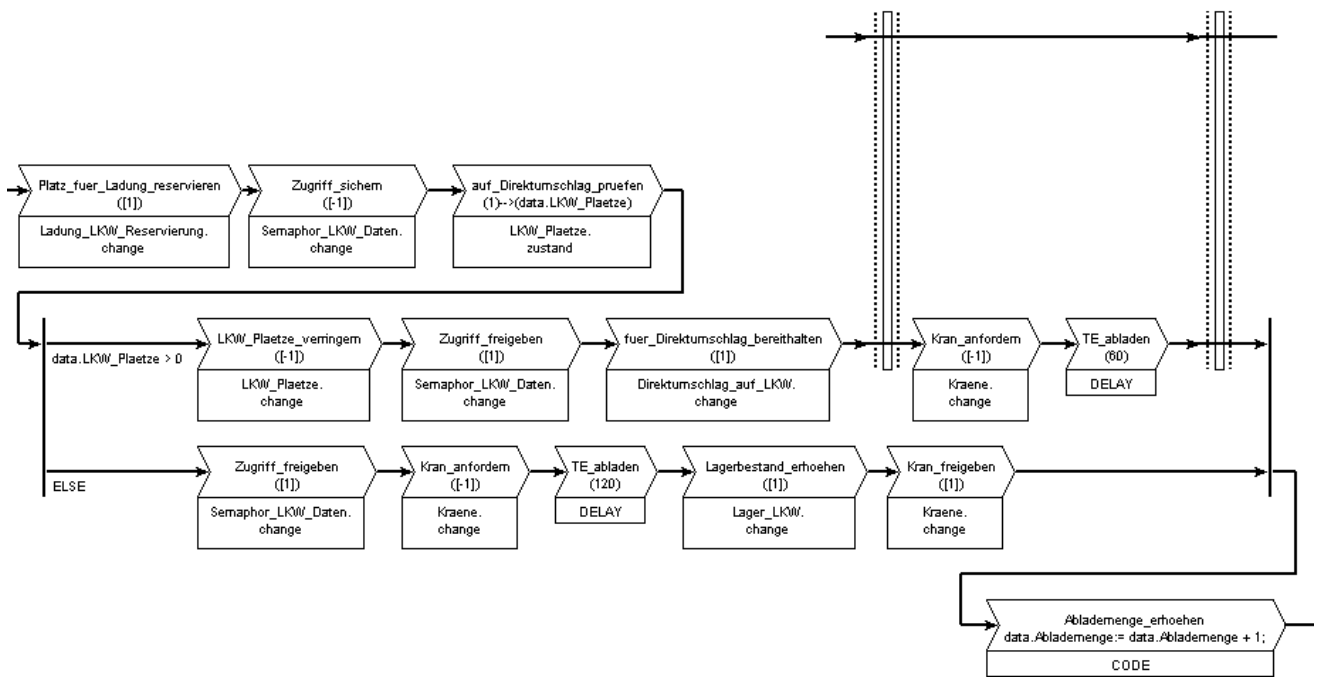


Abb. 22: Abladen einer TE von einem Zug (Ausschnitt aus den Prozessketten umschlagen_Zug (unten) und umschlagen_LKW (oben)).

Nach dem Abladen wird sowohl beim Direktumschlag als auch beim Zwischenlagern die Ablademenge verringert (vgl. Abb. 22).

Ist anschließend die Anzahl freier Plätze auf dem Zug (Variable *Freiraum*) kleiner als die Menge, die noch auf den Zug geladen werden muss ($data.TE_von_LKW - data.Auflademenge$ ergibt die noch aufzuladende Menge), so muss der sich durch das Abladen zusätzlich ergebende freie Raum auf dem Zug als für die Beladung zur Verfügung stehend gezählt werden (durch Veränderung der Variablen *Freiraum*, siehe Abb. 23). Gilt dabei im Aufladebereich für den Zug aktuell eine Ladesperre wegen fehlenden Freiraums, so ist die lokale Sperrvariable *Ladesperre* auf *False* zu setzen und die *Freiraum_Kontrolle* um ein Element zu vergrößern (vgl. Abb. 23), damit die Wartesituation im Aufladebereich aufgehoben werden kann (siehe Beschreibung des Aufladens). Ferner wird die Anzahl der *Zug_Plaetze* erhöht (vgl. Abb. 23), damit die LKW-Prozesse über diesen freien Raum verfügen können.

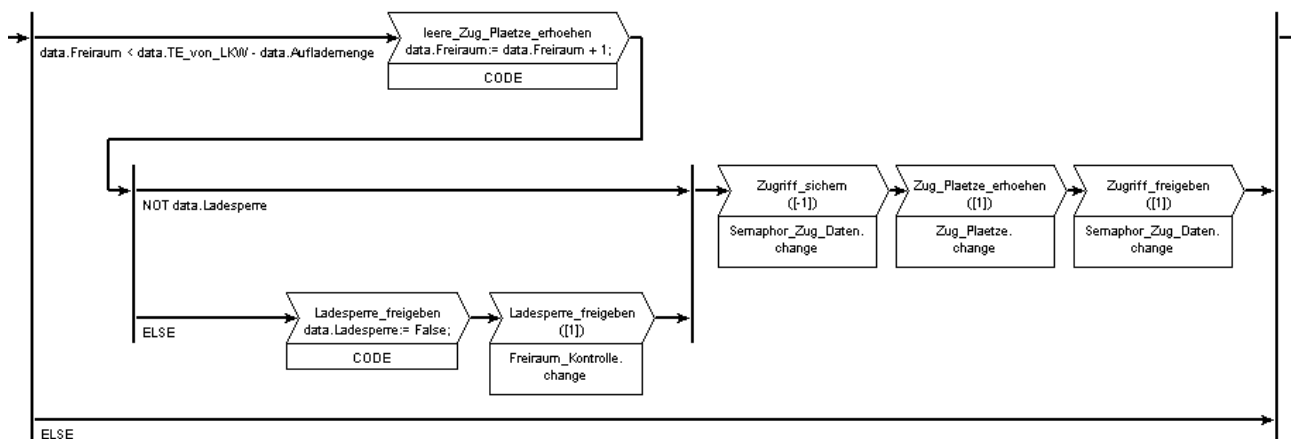


Abb. 23: Erhöhung des Freiraums und Freigabe der Ladesperre (Ausschnitt aus der Prozesskette umschlagen_Zug).

Ist nach einem Abladevorgang ein weiterer Abladevorgang nötig, so wird an den Anfang der Abladeschleife gesprungen. Ist nichts mehr von diesem Zug abzuladen, wird, nachdem auch die Beladung abgeschlossen ist, die Ausführung des Dienstes `umschlagen_Zug` beendet.

PK `umschlagen_LKW`

Auf der Prozesskette `umschlagen_LKW` entspricht jeder Prozess je einem LKW, der 2 Transporteinheiten transportieren kann (jedoch nicht muss, da auch teil- und unbeladene LKW zulässig sind).

Modellierung des Abladevorgangs

Falls etwas abzuladen ist, wird eine Abladeschleife (je Durchgang für jeweils eine Transporteinheit) durchlaufen (vgl. Abb. 18).

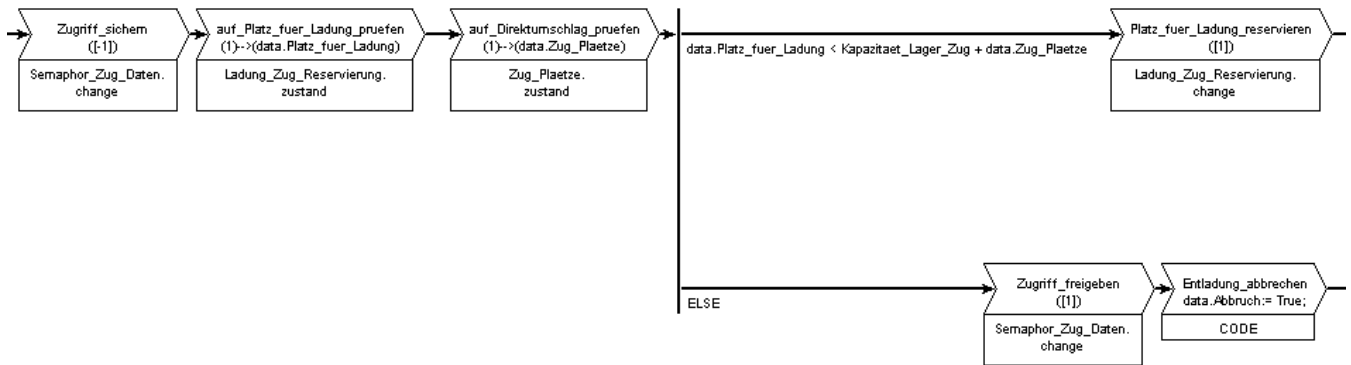


Abb. 24: Prüfung auf Platz für Ladung und auf Direktumschlag sowie evtl. Abbruch des Entladevorgangs (Ausschnitt aus der Prozesskette `umschlagen_LKW`).

Innerhalb der Schleife wird zunächst auf freien Platz auf einem Zug bzw. auf der Abstellspur getestet (vgl. Abb. 24). Gibt es keinen freien Platz mehr wird das Abladen abgebrochen, alle nicht bis dahin abgeladenen Transporteinheiten verbleiben auf dem LKW, der dem jeweiligen Prozess entspricht. Kann jedoch ein Abladen erfolgen, so wird Platz für die Ladung auf dem Zug bzw. auf der Abstellspur reserviert (PKE `Platz_fuer_Ladung_reservieren`).

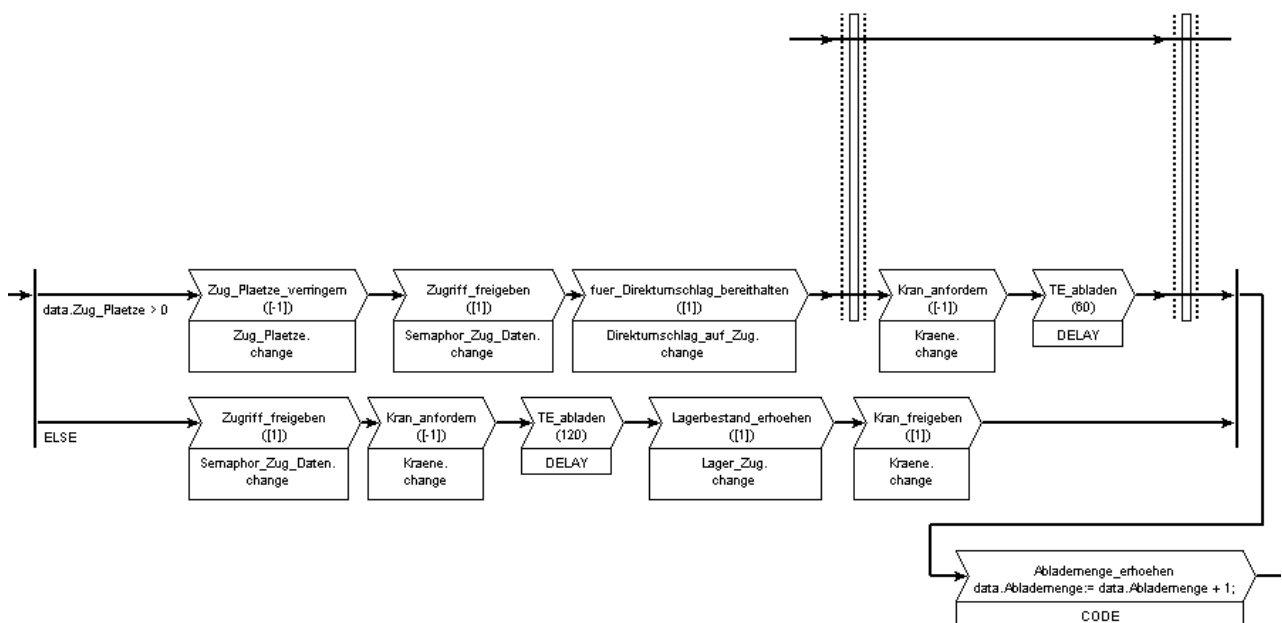


Abb. 25: Abladen einer TE von einem LKW (Ausschnitt aus den Prozessketten `umschlagen_Zug` (oben) und `umschlagen_LKW` (unten)).

Anschließend wird, falls kein Direktumschlag erfolgen kann, analog zum Abladen beim Zug, ein Kran angefordert, die Transporteinheit abgeladen und als für die Aufladung auf einen Zug zwischengelagert registriert (PKE Lagerbestand_erhoehen) sowie der Kran freigegeben (vgl. Abb. 25).

Im Fall des Direktumschlags wird ein Zugplatz belegt (die Belegungsmöglichkeit ist durch die vorhergehende Prüfung garantiert) und die Transporteinheit zu denjenigen addiert, die für Direktumschläge auf Züge warten (PKE fuer_Direktumschlag_bereithalten). Nachdem ein Zug zusammen mit dem LKW den Direktumschlag beginnen kann (die Gleichzeitigkeit ist durch den ersten PK-Konnektor in Abb. 25 modelliert), fordert der LKW einen Kran an und lädt die Transporteinheit ab. Danach wird die Kontrolle über den Direktumschlag an den Zug abgegeben (die dafür notwendige Synchronisation ist durch den zweiten PK-Konnektor in Abb. 25 modelliert).

Sowohl beim Zwischenlagern als auch beim Direktumschlag wird die Ablademenge erhöht (siehe Abb. 25) und wieder an den Anfang der Abladeschleife gesprungen, falls noch etwas abzuladen ist und kein Abbruch des Abladens durchgeführt wurde.

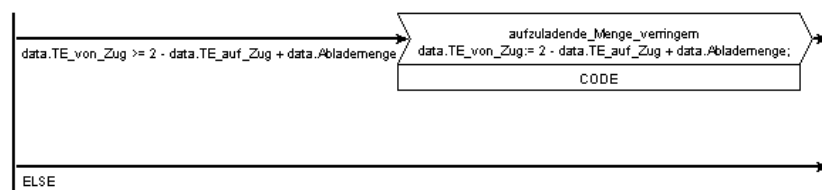


Abb. 26: Anpassen der aufzuladenden Menge (Ausschnitt aus der Prozesskette umschlagen_LKW).

Modellierung des Aufladevorgangs

Nach dem Verlassen der Abladeschleife wird zunächst die aufzuladende Menge angepasst (siehe Abb. 26). Dies ist für den Fall notwendig, dass das Abladen abgebrochen werden musste und dadurch die aufzuladende Menge nicht vollständig auf den LKW verbracht werden kann ($data.TE_auf_Zug - data.Ablademenge$ gibt dabei die wegen Abbruchs verbliebene Menge an, 2 ist die Kapazität eines LKW, $2 - data.TE_auf_Zug + data.Ablademenge$ ist somit die Menge, die höchstens aufgeladen werden könnte).

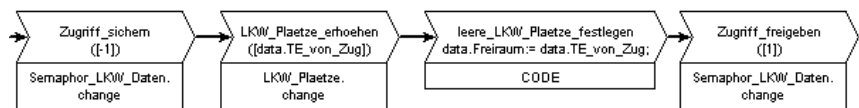


Abb. 27: Festlegen des für aufzuladende Transporteinheiten auf dem LKW zur Verfügung stehenden Platzes (Ausschnitt aus der Prozesskette umschlagen_LKW).

Verbleibt noch eine aufzuladende Menge, so werden zur Benachrichtigung abladewilliger Zug-Prozesse die LKW_Plaetze um den so ermittelte Freiraum erhöht (PKE LKW_Plaetze_erhoehen) und der aktuelle Freiraum des LKW in der prozesslokalen Variablen Freiraum festgehalten (vgl. Abb. 27).

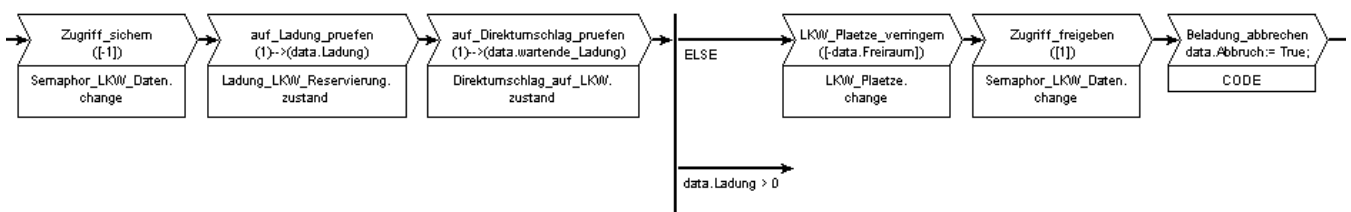


Abb. 28: Prüfung auf Ladung und Direktumschlag sowie evtl. Abbruch des Beladevorgangs (Ausschnitt aus der Prozesskette umschlagen_LKW).

Anschließend wird eine Aufladeschleife durchlaufen (vgl. Abb. 18), bei der jeder Durchlauf dem Aufladen einer Transporteinheit entspricht. Innerhalb der Schleife wird zunächst geprüft, ob Ladung auf der

Abstellspur oder im Direktumschlag auf einen LKW wartet (zur Motivation des Bausteins Ladung_LKW_Reservierung siehe Abschnitt über die Zugbeladung). Ist dies nicht der Fall, werden alle freien LKW-Plätze des LKW-Prozesses gelöscht (PKE LKW_Plaetze_verringern) und das Aufladen abgebrochen (siehe Abb. 28).

Falls etwas aufgeladen werden kann, wird die Auflademenge erhöht, Ladung reserviert und der aktuelle Freiraum des LKW verringert (vgl. Abb. 29).

Bei Entnahme von der Abstellspur werden die Anzahl der LKW-Plätze um 1 verringert (da dieser durch die folgende Beladung belegt wird), ein Kran wird angefordert, die Transporteinheit aufgeladen und anschließend der Kran wieder zurückgegeben (siehe Abb. 29).

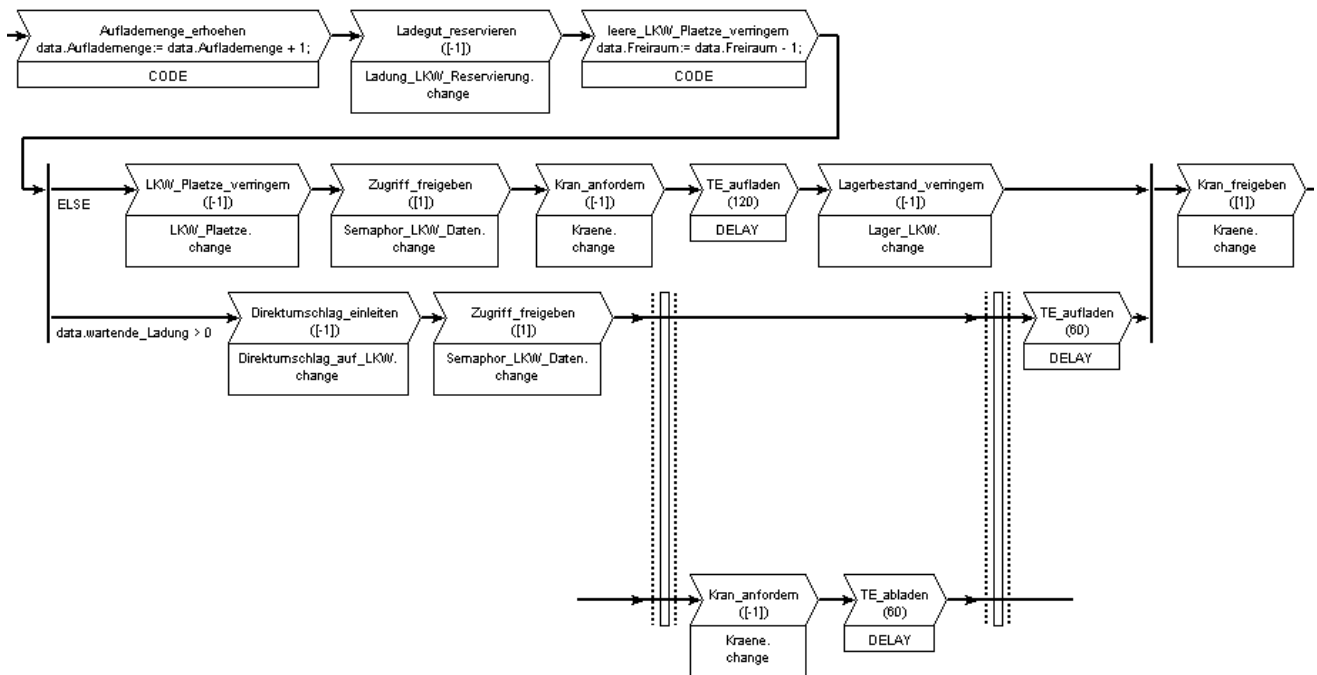


Abb. 29: Beladen eines LKW mit einer TE (Ausschnitt aus den Prozessketten umschlagen_LKW (oben) und umschlagen_Zug (unten)).

Bei Durchführung eines Direktumschlags wird die Anzahl der LKW-Plätze nicht durch den LKW-Prozess verringert (dafür ist der zugehörige Zug-Prozess zuständig). Jedoch wird eine direkt umzuschlagende Transporteinheit reserviert und dadurch der Direktumschlag eingeleitet (PKE Direktumschlag_einleiten). Ist ein entladender Zug-Prozess gemeinsam mit dem LKW-Prozess zum Direktumschlag bereit (durch den ersten PK-Konnektor in Abb. 29 modelliert), so fordert der Zug-Prozess einen Kran an und lädt die Transporteinheit ab. Anschließend (durch den zweiten PK-Konnektor in Abb. 29 modelliert) wird durch den LKW-Prozess die Transporteinheit auf den LKW aufgeladen und der Kran zurückgegeben. Eine Blockierung des Krans beim Aufladen durch einen LKW-Prozess ist nicht möglich, da stets garantiert ist, dass eine Ladung vorhanden ist, und Kräne nur während des Be- bzw. Entladens durch Zug- bzw. LKW-Prozesse gehalten werden. Hierdurch werden Deadlocks vermieden.

Nach erfolgter Beladung des LKW wird wieder an den Anfang der Aufladeschleife gesprungen, falls noch etwas aufzuladen ist und kein Abbruch durchgeführt wurde. Im anderen Fall ist die Beladung abgeschlossen und die Ausführung des Dienstes umschlagen_LKW beendet.

2.5 Abzuleitende Modellierungsgrundsätze

Aus der Modellierung des KV-Terminals lassen sich allgemeine Grundsätze ableiten, die man bei der Erstellung eines ProC/B-Modells berücksichtigen sollte, um Deadlocks zu vermeiden.

Bei alternativen Abläufen sollten die Stellen, an denen auf ein Ereignis gewartet wird, möglichst nicht über die Alternativen verteilt sein. Wird das Ereignis ebenfalls hinsichtlich dieser Alternativen unterschieden, so

kann es anderenfalls dazu kommen, dass ein solches Ereignis eine Wartesituation nicht auflösen kann (vgl. Motivation der Verwendung der Bausteine Ladung_Zug_Reservierung und Ladung_LKW_Reservierung im Kapitel 2.4.5.2.3).

Bei der Benutzung von Ressourcen sollten die Ressourcen möglichst nur dann von einem Prozess gehalten werden, wenn sie benötigt werden, und nicht, wenn der Prozess in eine Wartesituation geraten kann. Das Warten mit gehaltenen Ressourcen könnte dazu führen, dass andere Prozesse wegen nicht erfüllbaren Ressourcenanforderungen warten müssen und schlimmstenfalls in einen Deadlock führen, wenn der ressourcenhaltende Prozess erst weiterlaufen kann, nachdem auch die ressourcenanfordernden Prozesse weiterlaufen können. Als Beispiel sei auf die Krananforderungen im KV_Umschlag (siehe Kapitel 2.4.5.2.3) verwiesen, die erst erfolgen, wenn sicher ist, dass die Kräne auch unmittelbar danach Be- oder Entladungstätigkeiten durchführen können.

Müssen dennoch der Realität entsprechend (z. B. bei Parkplätzen) Ressourcen in Wartesituationen gehalten werden, so wird bei stochastischen Modellen ein Abweisungsmechanismus benötigt, der unter bestimmten Umständen Prozesse nicht in eine Wartesituation geraten lässt, sondern sie vorher die entsprechende Einheit verlassen lässt. Als Beispiel sei hier das Abbrechen von Be- oder Entladungsvorgängen bei den LKW im KV_Umschlag (siehe Kapitel 2.4.5.2.3) genannt, das dafür sorgt, dass z. B. auf Ladung wartende LKW nicht die Umschlagsplätze belegen, die jene LKW bräuchten, die die Wartesituation durch Entladungen beheben könnten. Durch den eingebauten LKW-Abweisungsmechanismus können aufladende Züge abgefertigt werden und so Platz für andere Ladung bringende Züge machen, wodurch auch wieder die Beladung von LKW möglich wäre.

Selbst wenn keine Zufallsverteilungen für das Quellverhalten herangezogen werden, sondern Traces aus Beobachtungen, ist es in komplexen Modellen wie dem GVZ-Modell nur schwerlich abzuschätzen, ob nicht dennoch Deadlocksituationen eintreten können. Somit wäre hier eine Missachtung obiger Regeln zwar möglich, es würde jedoch eine intensive Analyse der Eingangsdaten erfordern.

3 Ziele der Simulation und Vorgehensweise

3.1 Allgemein

Die Modellierung des GVZ im ProC/B-Modellierungsparadigma dient der Bestimmung der auf die Leistungsfähigkeit und Wirtschaftlichkeit Einfluss nehmenden Layout- und Betriebsparameter in einem GVZ. Das Modell soll dem Planer eines neuen GVZ Planungsrichtlinien zur Dimensionierung der Ressourcen und dem Betreiber eines bestehenden GVZ Ansätze zur Optimierung der Ressourcen und Prozesse zur Verfügung stellen. Die in dieser Hinsicht signifikanten logistischen Kenngrößen sind Durchsatz, Durchlaufzeit, Auslastungsgrad, Bestand und anfallende Kosten. Diese Kenngrößen befinden sich in einer Abhängigkeit zu den einzelnen Betriebsparametern der im Modell existierenden Ressourcen, so dass Zielgrößen für die Durchführung der Simulation formuliert werden können. Diese Zielgrößen sind für die verwendeten Module jeweils einzeln zu definieren, z. B. der Auslastungsgrad der Rampen in der Stückgutumschlaghalle, die Durchlaufzeit eines LKW oder der Bestand auf einer Abstellspur im KV-Terminal.

Auf das Modell bezogen lassen sich die wesentlichen Zielgrößen folgendermaßen beschreiben:

- Bestimmung der Durchlaufzeit von Leistungsobjekten
- Dimensionierung der Ressourcen
- Bestimmung von kritischen Zeitpunkten
- Tagesgangkennlinien aller Ressourcen

Als Basis der Simulation und um die formulierten Ziele zu quantifizieren wird ein Experimentplan in Anlehnung an die VDI-Richtlinie 3633 [VDI95] erstellt, nach dem beim Experimentdurchlauf vorgegangen wird. Bei einer Simulation sind zum einen das System selbst und zum anderen die Systemlast variabel.

Das System lässt sich in den durch Parameter einstellbaren Ressourcen sowie durch die Implementierung unterschiedlicher Strategien ändern. Die Systemlast kann in ihren Parametern dem Verwendungszweck entsprechend frei dimensioniert werden. Aus der Kombination dieser beiden Variationsmöglichkeiten ergeben sich für den Experimentplan insgesamt vier grundlegend verschiedene Anwendungsfälle, die an dem Modell durchgeführt werden können und Simulationsergebnisse zu jeweils unterschiedlichen Fragestellungen liefern.

Im ersten Anwendungsfall des Experimentplans wird die Simulation zunächst bei unverändertem Modell und konstant parametrierter Systemlast durchgeführt. Mit diesem Vorgehen kann eine Systemanalyse und -bewertung vorgenommen werden, wodurch die Funktionalität des Modells sichergestellt und eine grundsätzliche Aussage über die Performance des Modells getroffen werden kann. Durch das Durchführen dieses ersten Anwendungsfalles wird sichergestellt, dass im GVZ-Modell keine Deadlock-Situationen auftreten und das GVZ-Modell grundsätzlich lauffähig ist.

Im zweiten Fall werden die Parameter der Ressourcen bei konstant parametrierter Systemlast variiert, d. h. die Ressourcen werden sukzessive geändert, so dass die Kombination der Parameter im Modell bei der bestehenden Systemlast die Zielkriterien bestmöglich erfüllt. Dieses Vorgehen dient der Abstimmung der Zahl der Ressourcen im GVZ an eine vorgegebene Ankunftsverteilung, so dass für die Leistungsfähigkeit des GVZ hinsichtlich der formulierten Ziele ein Optimum in der Dimensionierung der Ressourcen erreicht wird. Mit dieser Vorgehensweise lässt sich eine Optimierung der Anzahl der Arbeitsmittel und Ressourcen bei einem existierenden GVZ durchführen.

Der dritte Anwendungsfall beinhaltet die Variation der Systemlast bei konstanten Einstellungen der Ressourcen. Dabei wird die Systemlast bis zu dem Moment erhöht, an dem das System die Last nicht mehr verarbeiten kann und nicht stationär wird. Zu diesem Zeitpunkt übersteigt der Ankunftsstrom von Leistungsobjekten die partielle Grenzleistung einzelner Ressourcen oder es werden Bestände in den Lagern nicht mehr abgebaut. Dieser Anwendungsfall ermöglicht bei einem bestehenden GVZ die Bestimmung der Leistungsgrenzen der einzelnen Ressourcen.

Der vierte Anwendungsfall umfasst die Variation sowohl des Systems als auch der Systemlast. In diesem Teil des Experimentierplans sollen allgemeingültige Aussagen über typische Systemstrukturen getroffen werden. Die Variation des Systems und der Systemlast erfolgt unter statistischen und heuristischen Gesichtspunkten. Auf Seiten des Modells werden in mehreren aufeinander folgenden Simulationsschritten ausgewählte Ressourcen wie Umschlagpersonal, Administrationspersonal, Rampen, Park-, Warte- und Umschlagplätze, Lagerflächen, Personal am KV- Terminal und Anzahl der Kräne unter Anwendung verschiedener Strategien sukzessive verändert. Auf Seiten der Systemlast besteht die Variation zum einen in der Verwendung verschiedener Volumina der Eingangsgrößen und zum anderen in der Anwendung unterschiedlicher zeitlich diskretisierter Verteilungen des Ankunftsverhaltens der Verkehrsträger.

3.2 Ziele für die SUH

Mit Hilfe der Simulation kann die Fläche der Stückgutumschlaghalle aufgrund der Lagerverlaufskennlinie, der Anzahl der nötigen Rampen, der Größe der innenliegenden Verkehrswege und der nötigen innenseitigen Pufferflächen bestimmt werden.

Die Hallenfläche im Außenbereich ist abhängig von der Halleninnenfläche, den Verkehrswegen, dem Platzbedarf der Administration der Größe der Fläche für Park- und Warteplätze. Halleninnenseitige Flächen ergeben sich aus der Breite der Halle, die durch die Anzahl der Tore bestimmt ist, der Größe der Pufferfläche, der Art der Lagerung des Puffergutes und der dafür vorgesehen Verkehrswege.

Im Weiteren können die Durchlaufzeiten der Transportmittel vom Stückgutumschlaghalleneingang bis zur Einlagerung, die Durchlaufzeit vom Auslagern bis zum Verlassen der Stückgutumschlaghalle und der Bedarf an Ressourcen ermittelt werden.

3.3 Ziele für das KV-Terminal

Eine Simulation kann zunächst Aufschlüsse über die Auslastung der einzelnen Ressourcen geben, wobei die Anzahl der Mitarbeiter am Gate, die Auslastung der Parkplätze, die Auslastung der Verladespur, die

Auslastung der Kräne und die Belegung der Abstellspur von größtem Interesse sind. Dabei lässt sich bei der Abstellspur noch nach Auf- und Abladevorgängen sowie nach Herkunft der Ladeeinheiten, von LKW oder Zug, differenzieren. Unter Verwendung der Aus- und Einlagerzeitpunkte, lässt sich zudem eine Lagergangkennlinie der einzelnen Storages erstellen. Im Weiteren kann die Anzahl der Direktumschläge mit Hinterlegung der Richtung gemessen werden, woraus dann Schlussfolgerungen dahingehend gezogen werden können, ob und vor allem auch zu welchen Zeitpunkten auf den Tag bezogen, ein Verkehrsträger auf den anderen warten muss. Somit lassen sich Aussagen über den Zeitpunkt von Belastungsspitzen treffen, sowie über die sie verursachende Ressource. Durch Anpassen der Parameter der Ressourcen, z. B. durch Erhöhen oder Senken der Kapazitäten, an die herausgestellten kritischen Situationen lassen sich die Auswirkungen auf das System feststellen, und es ist somit möglich, ein System sukzessive zu optimieren. Hierbei liegt der Fokus bei der Optimierung des Gesamtsystems auf Minimierung der Durchlaufzeiten, dazu gehört auch Erhöhung der Direktumschläge, und Kapazitäten bei Einhaltung des geforderten Zeitfensters. Ein weiterer Aspekt ist der zeitliche Vergleich der Anzahl der anwesenden Verkehrsträger oder der umzuschlagenden Transporteinheiten mit der Auslastung der Kräne bei Belastungsspitzen, um Aussagen über eine eventuell notwendige gesteuerte Abstimmung der Ankunftszeiten der Verkehrsträger zur Reduzierung von Belastungsspitzen oder über den Einsatz von zusätzlichen Unstetigförderern bei Belastungsspitzen zu treffen.

4 Fazit

Mit Hilfe des ProC/B-Paradigmas konnte ein Modell erstellt werden, das komplexe Abläufe eines GVZ nachbildet und die Analyse verschiedener Varianten erlaubt. Die Lauffähigkeit des Modells konnte anhand erster Simulationsläufe sowohl mit stochastischen als auch mit der Realität entnommenen Daten festgestellt werden. Das Modell ist somit geeignet, die für die verschiedenen Simulationsläufe formulierten Ziele erreichen zu können. In diesen Simulationsläufen konnten erste Aussagen über die Betriebsparameter ausgewählter Ressourcen ermittelt werden, so dass bereits Schlussfolgerungen hinsichtlich der Dimensionierung einzelner Ressourcen getroffen werden konnten.

Darüber hinaus konnten bei der Modellierung Erkenntnisse über Grundsätze bei der Modellerstellung gewonnen werden.

Weiterer Handlungsbedarf besteht im Weiteren in der systematischen Durchführung der in Kapitel 3 beschriebenen Vorgehensweise zur Bestimmung allgemeingültiger Aussagen zur Dimensionierung eines GVZ.

5 Literatur

- [BBF+02] F. Bause, H. Beilner, M. Fischer, P. Kemper, M. Völker: The Proc/B Toolset for the Modelling and Analysis of Process Chains. In: Field, T.; Harrison, P.-G.; Bradley, J.; Harder, U. (Hrsg.): Computer Performance Evaluation Modelling Techniques and Tools. 12th International Conference on Modelling Techniques and Tools (Performance Tools 2002), London, UK, 14.-17. April, LNCS 2324, Springer, Berlin Heidelberg New York, 2002, S. 51-70.
- [BBF+03] F. Bause, J. Bernhard, T. Fender, K. Hömberg, D. Jodin, M. Völker, S. Wenzel: Standardisierte Beschreibung von Eingangsdaten für die Simulation auf Basis des Prozesskettenparadigmas, SFB559, Bericht 03004, Dortmund, 2003
- [HIT93] M. Büttner (ed.), B. Fricke, O. Klößen, S. Nolte, H. Stahl, N. Weißenberg. Hi-Slang Reference Manual for the Hierarchical Evaluation Tool HIT. Universität Dortmund, Informatik IV, 1993.
- [Ku95] A. Kuhn. Prozessketten in der Logistik, Entwicklungstrends und Umsetzungsstrategien. Verlag Praxiswissen, Dortmund, 1995.
- [VDI95] VDI 3633, Blatt 3: Verein Deutscher Ingenieure, Düsseldorf, Simulation von Logistik-, Materialfluß- und Produktionssystemen, VDI-Gesellschaft Fördertechnik, Materialfluß, Logistik, Beuth Verlag, Berlin, 1995.