

Otto-von-Guericke-Universität Magdeburg



OTTO VON GUERICKE  
UNIVERSITÄT  
MAGDEBURG

INF

FAKULTÄT FÜR  
INFORMATIK

Fakultät für Informatik  
Institut für Technische und Betriebliche Informationssysteme

## Bachelorarbeit

### **Konzeption und Entwicklung einer automatisierten Schnittstelle für Warenwirtschafts- und Versandssysteme von Onlineshops**

Verfasser:

Steven Ladewig

03. Februar 2012

Betreuer:

Prof. Dr. rer. nat. habil. Gunter Saake,  
Dipl.-Inform. Andreas Lübcke

Universität Magdeburg  
Fakultät für Informatik  
Postfach 4120, D-39016 Magdeburg  
Germany

**Ladewig, Steven:**

*Konzeption und Entwicklung einer automatisierten Schnittstelle für Warenwirtschafts- und Versandsysteme von Onlineshops*

Bachelorarbeit, Otto-von-Guericke-Universität  
Magdeburg, 2012.

## Danksagung

Ich möchte mich vielmals für die Betreuung und die Möglichkeit zur Verfassung dieser Arbeit bei Prof. Dr. rer. nat. habil. Gunter Saake und bei Dipl. Inf. Andreas Lübcke bedanken. Auch möchte ich mich bei allen Mitarbeitern der Qubeat GmbH für die Themenstellung dieses Werkes bedanken. Besonderer Dank gilt hierbei Anja Altner, Tim Müller, Tanyel Savci und Christian Heuberger, die maßgeblich an der Gestaltung der Aufgabenstellung beteiligt waren. Weiterer Dank gilt insbesondere meiner Familie sowie meinen Freunden für das wiederholte Korrekturlesen.

---

---

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>ii</b>
<b>Abbildungsverzeichnis</b>	<b>iv</b>
<b>Tabellenverzeichnis</b>	<b>v</b>
<b>Verzeichnis der Abkürzungen</b>	<b>vi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Ausgangslage der Versandbearbeitung . . . . .	2
1.2 Zielsetzung . . . . .	4
<b>2 Theoretische Grundlagen</b>	<b>5</b>
2.1 Begriffserklärungen . . . . .	5
2.1.1 Objektorientierte Programmierung . . . . .	5
2.1.2 Abstrakte Klassen und Interfaces . . . . .	5
2.1.3 Schnittstelle . . . . .	5
2.2 Packproblem . . . . .	6
2.2.1 Brute-Force-Methode . . . . .	6
2.2.2 Genetische Algorithmen . . . . .	6
<b>3 Konzeption</b>	<b>10</b>
3.1 Systemarchitektur . . . . .	10
3.2 Einordnung des Versandsystems . . . . .	10
3.3 Datenbeschaffenheit . . . . .	12
3.3.1 Bestelldaten . . . . .	12
3.3.2 Lagerdaten . . . . .	12

---

---

3.4	Klassifikation von Bestellungen . . . . .	13
3.5	Versandschnittstellen . . . . .	14
<b>4</b>	<b>Implementierung</b>	<b>17</b>
4.1	Klassifikation von Paketen . . . . .	17
4.1.1	Vorverarbeitung . . . . .	17
4.1.2	Erkennung . . . . .	21
4.2	Benutzeroberfläche . . . . .	24
4.3	Integrierbarkeit . . . . .	25
<b>5</b>	<b>Evaluation</b>	<b>27</b>
5.1	Berechnungszeit . . . . .	27
5.2	Lernsystem . . . . .	29
5.3	Speicherauslastung . . . . .	29
5.4	Arbeitsersparnis . . . . .	30
<b>6</b>	<b>Vergleich mit anderen Versandlösungen</b>	<b>31</b>
6.1	DHL Intraship . . . . .	31
6.2	HVS32 . . . . .	32
<b>7</b>	<b>Zusammenfassung</b>	<b>34</b>
	<b>Literaturverzeichnis</b>	<b>35</b>

---

---

# Abbildungsverzeichnis

1.1	Grundsystem . . . . .	2
1.2	Auswahl der Lieferdienste in <i>DreamRobot</i> . . . . .	3
1.3	Auswahl des Exportformats . . . . .	4
2.1	Crossover . . . . .	9
2.2	Mutation . . . . .	9
3.1	Vorhandene Softwarestruktur . . . . .	11
3.2	Einordnung . . . . .	11
4.1	Ablauf der Klassifikation . . . . .	18
4.2	Lageraufteilungsmöglichkeiten . . . . .	20
4.3	Gegenüberstellung Brief- und Paketsendung . . . . .	21
4.4	Versandvorschläge . . . . .	24

---

---

# Tabellenverzeichnis

3.1	Maximalmaße ausgewählter Lieferdienste . . . . .	13
4.1	Filterkriterien . . . . .	19
5.1	Berechnungszeiten für je 50 Bestellungen . . . . .	28
5.2	Erkennungen über das Lernsystem . . . . .	29
5.3	Speicherauslastung . . . . .	29
5.4	Klicks zur vollständigen Verarbeitung von 50 Bestellungen . . . . .	30
6.1	Gegenüberstellung der Systeme . . . . .	31

---

---

# Verzeichnis der Abkürzungen

<b>API</b>	Application Programming Interface
<b>BITKOM</b>	Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.
<b>CSS</b>	Cascading Style Sheets
<b>CSV</b>	Comma Separated Values
<b>cURL</b>	Client for URLs
<b>GA</b>	genetischer Algorithmus
<b>GHz</b>	Gigahertz
<b>GLS</b>	General Logistics Systems
<b>HVS32</b>	Heidler Versandsystem 32 Bit
<b>ID</b>	Identifikator
<b>ISO</b>	Internationale Organisation für Normung
<b>IT</b>	Informationstechnik
<b>ODBC</b>	Open Database Connectivity
<b>PC</b>	Personal Computer
<b>PDF</b>	Portable Document Format
<b>PHP</b>	PHP: Hypertext Preprocessor / Personal Home Page Tools
<b>RAM</b>	Random Access Memory
<b>URL</b>	Uniform Resource Locator



# Kapitel 1

## Einleitung

Der Anteil an Internetnutzern über 14 Jahren liegt bei der deutschen Bevölkerung einer Umfrage des BITKOM zufolge bei etwa 72 Prozent [BHA11]. Das entspricht circa 60 Millionen Nutzern. Davon haben 85 Prozent schon einmal Waren oder Dienstleistungen online gekauft [BHA11]. Der Online-Handel ist demnach ein Markt mit einem großen Potential. Durch zahlreiche kostengünstige Shop-Systeme, auf die zurückgegriffen werden kann, ist der Onlinemarkt für kleine und größere Händler interessant. Während größere Händler im Regelfall häufig auf Eigenentwicklungen setzen, nutzen kleinere Unternehmen aus Kostengründen eher kostenlose Systeme beziehungsweise Kauf- und Miet-Shops.

Während sich der Einkauf für den Kunden durch One-Site-Checkouts<sup>1</sup> und andere Komfortfunktionen, wie zum Beispiel strukturierte Übersichten, bei den gekauften, gemieteten und kostenlosen Systemen immer weiter vereinfacht, gibt es in der Bestellabarbeitung oftmals Bereiche, die große Lücken aufweisen. Ein Beispiel hierfür sind fehlende Anbindungen an Online-Plattformen wie Amazon und eBay, über die Artikel auch auf solchen Plattformen angeboten werden können. 60 Prozent der Unternehmen setzen daher auf Warenwirtschaftssysteme, die bereits Anbindungen an verschiedene Portale beinhalten [WSW<sup>+</sup>11].

Weiterhin nutzen über 40 Prozent der Onlinehändler mehrere Versanddienstleister, um Kosten einzusparen [WSW<sup>+</sup>11]. Während einige Anbieter dem Kunden die dadurch auftretende Ersparnis weiterreichen, indem der Kunde das Versandunternehmen für seine Bestellung selbst auswählen kann, gibt es auch Anbieter, die dem Kunden einen Pauschalpreis auferlegen, um so die Versandart selbst wählen zu können. Um die Versandkosten zu minimieren sind diese Händler auf eine Optimierung angewiesen. Die bereits erhältliche Shop-Software bietet im Regelfall jedoch keine Möglichkeit den Händler oder den Kunden bei der Paketkostenoptimierung zu unterstützen. In diesem Fall basiert die Entscheidung, wie ein Päckchen verschickt wird, auf Erfahrungswerten des zuständigen Mitarbeiters. Diese Einteilung führt insbesondere bei neuen Produkten oft zu Fehlern, die durch ein automatisches Versandsystem vermieden werden können. Ein weiterer Vorteil bei einem automatischen Versandsystem ist die Zeitersparnis durch den Wegfall der manuellen Paketeinteilung.

Um die Vorteile sowie entstehende Nachteile gegenüber der manuellen Einteilung, aber

---

<sup>1</sup>Bezahlvorgang eines Onlineshops ist auf einer Seite zusammengefasst.

auch gegenüber anderen Versandlösungen zu verdeutlichen wird nachfolgendes Fallbeispiel dienen.

## 1.1 Ausgangslage der Versandbearbeitung

Im begleitenden Anwendungsfall liegt ein xt:commerce 3.04 Shopsystem<sup>2</sup>, sowie das Warenwirtschaftssystem *DreamRobot* als vorhandene Softwarelösung vor.

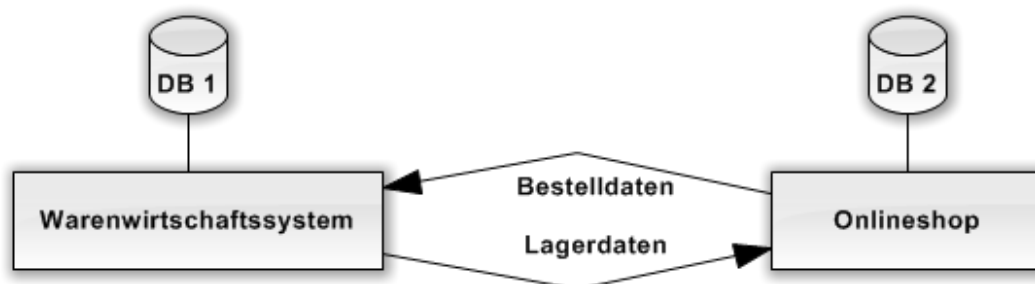


Abbildung 1.1: Grundsystem

Alle über den Shop eingehenden Bestellungen werden direkt an das Warenwirtschaftssystem *DreamRobot* übertragen. Dieses übernimmt die gesamte Abwicklung der Bestellungen. Der Shop dient lediglich als Schnittstelle zum Benutzer (siehe Abbildung 1.1). In der Shopoberfläche hat der Kunde die Auswahl, ob die Ware zugesandt werden soll oder ob er sie selbst abholt. Das Versandunternehmen hingegen kann nicht vom Kunden gewählt werden.

Für den Warenversand erhält ein Mitarbeiter eine umfangreiche Auswahlliste durch das Warenwirtschaftssystem. Die Auswahlliste enthält alle für diesen Tag zu versendenden Bestellungen. Auf der Basis von Erfahrungswerten wird für die Bestellungen eine von zahlreichen Versandarten ausgewählt, die das Warenwirtschaftssystem bereits anbietet. Wie in der Abbildung 1.2 zu sehen ist, wird in *DreamRobot* eine tabellarische Form mit vierzehn Radiobuttons pro Bestellung zur Darstellung genutzt.

In der Liste werden die Datensätze manuell entsprechend den Lagerorten der enthaltenen Produkte markiert. Im Fallbeispiel existieren zwei verschiedene Lagerorte<sup>3</sup>. Auf Grund der vorhandenen IT-Struktur müssen die Artikel, die vom Lagerort B versendet werden, an einem separaten Rechner bearbeitet werden, da nur dort eine Drucker-Verbindung zu Lagerort B besteht.

Für beide Lagerorte wird bei jeder Bestellung festgelegt, mit welchem Lieferunternehmen sie gesendet werden soll und wie viele Etiketten für den Versand nötig sind. Es ist von *DreamRobot* aus nicht möglich Teile einer Bestellung mit verschiedenen Lieferdiensten zu versenden, da die Auswahl hier, wie in der Darstellung 1.2 abgebildet,

<sup>2</sup>Ein freies Shopssystem von der xt:Commerce GmbH.

<sup>3</sup>Lagerort A: vor Ort; Lagerort B: Außenlager.



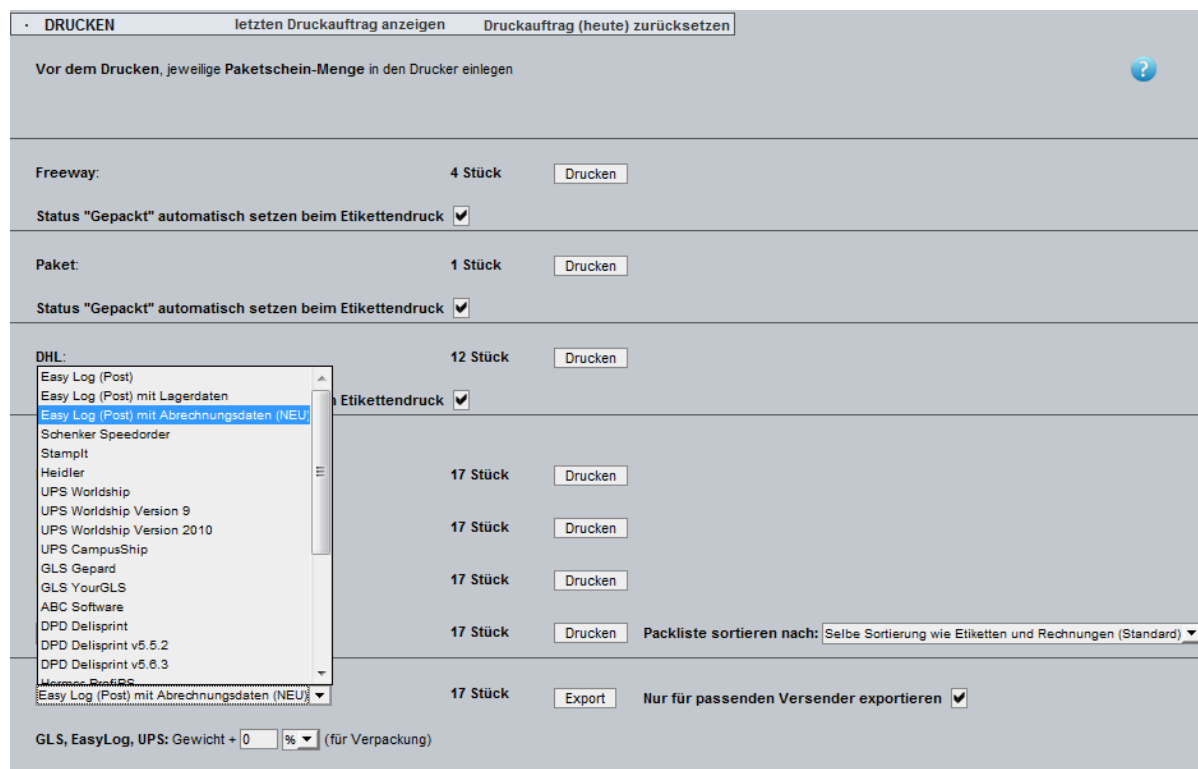


Abbildung 1.3: Auswahl des Exportformats

Auch für *Internetmarke*, das Onlineangebot der Deutschen Post, kann das Warenwirtschaftssystem *DreamRobot* eine CSV-Datei erstellen. In dieser Datei findet jedoch keine Unterteilung in die einzelnen Versandoptionen statt, sodass der Export für jede Versandart separat durchgeführt werden muss.

Im Gegensatz zu *Gepard* und *EasyLog* wird der Druckvorgang bei *Internetmarke* nicht automatisch gestartet. Der Benutzer bekommt hier PDF-Dateien mit den Etiketten zurückgeliefert. Der Druckvorgang muss hier manuell ausgelöst werden.

## 1.2 Zielsetzung

Um die zahlreichen Schritte, vom Eingang der Bestellung bis hin zum Druck der Etiketten, zu verkürzen, soll ein Versandsystem entwickelt werden. Dieses soll die manuelle Zuordnung zwischen den Bestellungen und den Versanddienstleistern weitestgehend automatisieren, beziehungsweise Teilschritte vereinfachen.

Um die Integrierbarkeit des Systems in andere Shops oder auch Warenwirtschaftssysteme zu gewährleisten, sollen keine Funktionsbibliotheken des gegebenen Onlineshops verwendet werden. Auf diese Weise kann das Versandsystem auch als Stand-Alone-Anwendung, also ohne die Integration in ein bereits vorhandenes System, verwendet werden.

Alle Daten, die das System benötigt, sollen über austauschbare Schnittstellen von verschiedenen Shop- und Warenwirtschaftssystemen importiert werden. Auch eine Erweiterung der Anwendung für andere Lieferdienste soll später einfach zu realisieren sein.

# Kapitel 2

## Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen für die späteren Ausführungen gelegt. Im ersten Abschnitt werden einige Begriffe, auf die sich diese Arbeit bezieht, erklärt. Um eine Automatisierung des Versands zu erzielen, ist es erforderlich, die Bestellungen, die im Warenwirtschaftssystem eingehen, den verschiedenen Versanddiensten zuzuordnen. Aus diesem Grund werden im zweiten Abschnitt dieses Kapitels zwei mögliche Verfahren zur Klassifizierung vorgestellt.

### 2.1 Begriffserklärungen

#### 2.1.1 Objektorientierte Programmierung

Die objektorientierte Programmierung ist ein Konzept, nach dem ein Programm umgesetzt werden kann [BB10, PH08]. Das Programm besteht hierbei aus Objekten, die bestimmte Probleme lösen [PH08]. Durch das Übermitteln von Nachrichten kommunizieren diese Objekte miteinander, um ein komplexeres Gesamtproblem zu lösen [BB10, PH08].

#### 2.1.2 Abstrakte Klassen und Interfaces

Bei abstrakten Klassen und Interfaces handelt es sich in der Programmierung um Klassen die bewusst unvollständig sind [BB10]. So enthalten abstrakte Klassen eine oder mehrere Methoden die zwar definiert aber nicht implementiert sind. Solche Methoden werden als abstrakt bezeichnet. Bei Interfaces handelt es sich um abstrakte Klassen, bei denen alle Methoden abstrakt sind [BB10].

Abstrakte Klassen als auch Interfaces werden in der objektorientierten Programmierung hauptsächlich verwendet, um zwischen zwei ähnlichen Klassen mit Hilfe einer solchen Oberklasse eine Beziehung herzustellen, mit der verschiedene Konzepte der Polymorphie und des dynamischen Bindens von Methoden genutzt werden können [BB10].

#### 2.1.3 Schnittstelle

Eine Schnittstelle (engl. Interface) bezeichnet in der Programmierung eine Sammlung von Programmfunktionen, die ein Softwaresystem anderen Anwendungen zur Anbindung

zur Verfügung stellt [Tho05].

Zu Abgrenzungszwecken bezeichnet das englische Wort “Interface“ in dieser Arbeit immer den Begriff aus der objektorientierten Programmierung, wohingegen das deutsche Wort “Schnittstelle“ für die allgemeine Bezeichnung von Programmierschnittstellen genutzt wird.

## 2.2 Packproblem

Um später ermitteln zu können, wie eine Bestellung zu verpacken ist, damit möglichst geringe Versandkosten entstehen, ist es nötig, die Abmessungen der potentiellen Pakete zu bestimmen. Dazu werden Ansätze aus der Packoptimierung verwendet. Damit wird die kleinstmögliche Packweise für eine Bestellung bestimmt. Es wird versucht eine gegebene Anzahl von Produkten, welche durch Quader repräsentiert werden, in einen möglichst kleinen Quader zu stapeln.

### 2.2.1 Brute-Force-Methode

Ein Ansatz zur Lösung des Packproblems nutzt die Brute-Force-Methode (Methode der rohen Gewalt). Hierbei werden alle denkbaren Paketzusammenstellungen einer Bestellung gebildet. Dieses Vorgehen “entspricht dem systematischen, vollständigen Durchsuchen des kompletten Zustandsraumes nach Lösungen“ [ZZZ07]. Der Brute-Force-Ansatz kann auf nahezu jedes Problem angewendet werden und führt bei endlichen Problemen theoretisch immer zum Ziel. Oftmals ist der zu durchsuchende Zustandsraum jedoch so groß, dass der Zeit- und Speicheraufwand die Kapazitäten aktueller PCs überschreitet [ZZZ07]. Dadurch können in der Praxis nicht alle Berechnungen durchgeführt werden.

Zur Selektion der optimalen Lösung werden alle Lösungen der Ergebnismenge anhand einer Bewertungsfunktion sortiert und die beste ausgegeben.

### 2.2.2 Genetische Algorithmen

Um die hohe Berechnungszeit, die das Brute-Force-Verfahren bei einer größeren Anzahl an Elementen mit sich bringt, zu vermindern, bieten sich alternativ genetische Algorithmen zur Lösung des Packproblems an. Die genetischen Algorithmen zählen zu den evolutionären Algorithmen [Lip06]. Bei den genetischen Algorithmen ist die Kodierung der Daten stärker durch die Genetik beeinflusst als bei den evolutionären Algorithmen [Pae06]. Im nachfolgenden Abschnitt soll der Basis GA (siehe Listing 2.1) vorgestellt werden.

Listing 2.1: Basis-GA

```
1 function basisGA(){
2   Erzeuge zufällige Startkombination aus kodierten Lösungen;
3   while( Abbruchbedingung nicht erfüllt){
4     Bewerte alle Individuen anhand einer Fitnessfunktion;
5     Matingpool ={};
```

```
6   Neue Population={};
7   while(Anzahl der Individuen im Matingpool kleiner als die Anzahl der vorhergehenden
      Individuen){
8     Kopiere ausgewähltes Individuum in Matingpool;
9   }
10  while(Neue Population unvollständig){
11    Wähle zwei Individuen aus dem Matingpool;
12    if(Individuen der Vorgängergeneration nicht identisch){
13      if(Wahrscheinlichkeit für Crossover){
14        Crossover durchführen;
15      }
16    }else{
17      Mutation durchführen;
18    }
19    Füge Individuen in Neue Population ein;
20  }
21  Alte Population = Neue Population;
22
23 }
24 Gib Ergebnisse aus;
25 }
```

## Grundlagen

In Anlehnung an die Biologie wird die kodierte Form einer Lösung für ein Problem im nachfolgenden als Individuum bezeichnet. Eine Menge von Individuen ist eine Population. Jedes Individuum hat Eigenschaften, diese werden als Erbgut bezeichnet.

Wie die Vererbungslehre, basieren auch die genetischen Algorithmen auf drei einfachen Prinzipien. Dazu zählen die Mutation, also die willkürliche Veränderung des Erbguts, die Neukombination der Erbinformationen (Crossover), sowie die Selektion der stärksten Individuen in einer Population [Lip06].

Eine Fitnessfunktion dient zur Bewertung der Individuen. Unter dem Begriff Fitness wird die Übereinstimmung einer Lösung mit einer je nach Problemstellung gegebenen Menge von Eigenschaften verstanden [Lip06]. Dieser Wert entspricht einem Kennwert, der die Qualität der Lösungen beschreibt [Lip06].

## Wahl einer Kodierung

Zunächst müssen die Ausgangsdaten für die weitere Verarbeitung kodiert werden. Bei der Wahl der Kodierung sollte darauf geachtet werden, dass eine Übertragung des Erbguts im Crossover-Prozess möglich ist. Eine kodierte Version einer Lösung wird als chromosomale Darstellung oder Genotyp bezeichnet.

## Initialisierung der Ausgangspopulation

Zu Beginn des Algorithmus wird eine Startpopulation zufällig generiert. Eine typische Anzahl an Individuen  $\mu$  in der Ausgangspopulation beträgt zwischen 30 und 500. Im Regelfall ist  $\mu$  geradzahlig, da dies die spätere Paarbildung vereinfacht [Nis97].

Die stochastisch ermittelten Lösungen der Startpopulation werden nun anhand der Fitnessfunktion bewertet [Nis97].

## Selektion und Replikation

Im nächsten Schritt werden aus der Ausgangspopulation  $\mu$  Individuen zufällig gezogen. Die Ziehung erfolgt mit Zurücklegen. Eine Mehrfachauswahl ist somit möglich. Die ausgewählten Individuen werden in einen so genannten *mating pool* übertragen und im Folgeschritt zur Erstellung der nächsten Generation an Lösungen genutzt [Nis97]. Für die Auswahl werden in der Literatur verschiedene Strategien vorgeschlagen.

**Zufällige Wahl** Bei der zufälligen Wahl werden die Individuen einer Population gleich wahrscheinlich ausgewählt. Hierbei wird die Suche nicht auf einen bestimmten Suchbereich konzentriert und die Vielfalt der Lösungen bleibt erhalten [Sau04].

**Proportionale Wahl** Als Alternative zur zufälligen Wahl bietet sich die proportionale Wahl an. Bei dieser Strategie ist die Wahrscheinlichkeit für die Auswahl eines Individuums proportional zu seiner Fitness. Für qualitativ bessere Lösungen besteht somit eine höhere Wahrscheinlichkeit, dass diese ausgewählt werden [Sau04].

**Platzabhängige Wahl** Die Selektion kann auch über die Platzierung der Fitness eines Individuums innerhalb der Population geschehen. Dadurch wird vermieden, dass zu viele Individuen einer Population identisch sind, denn das könnte dazu führen, dass der Entwicklungsprozess blockiert wird, also keine neuartigen Individuen entstehen können. Dies kann bei der proportionalen Wahl zum Beispiel geschehen, wenn ein Individuum eine weitaus höhere Fitness hat als alle anderen [Sau04].

## Bildung der Nachkommen

Auf Basis des *mating pools* wird nun die nächste Generation an Lösungen gebildet.

**Partnerwahl** Um neue Individuen zu bilden, werden immer paarweise Individuen aus dem *mating pool* gezogen. Die Wahrscheinlichkeit, dass ein Individuum gezogen wird, ist gleichverteilt. Die Ziehung erfolgt in diesem Schritt ohne Zurücklegen und endet erst, wenn der *mating pool* leer ist. Aus den Individuen-Paaren werden nachfolgend jeweils zwei Nachkommen gebildet [Nis97]. Die Größe der Population bleibt auf diese Weise in jedem Schritt erhalten.



**Crossover** Im Crossover-Schritt werden Merkmale der Eltern in der chromosomalen Darstellung vermischt. Ob ein solcher Austausch stattfindet wird wiederum stochastisch ermittelt [Lip06]. Eine typische Wahrscheinlichkeit für das Auftreten eines Tauschs liegt bei  $\geq 0,6$  [BBH<sup>+</sup>99, Nis97]. Findet kein Tausch statt, werden beide Elternteile in die nächste Generation übernommen [Nis97].

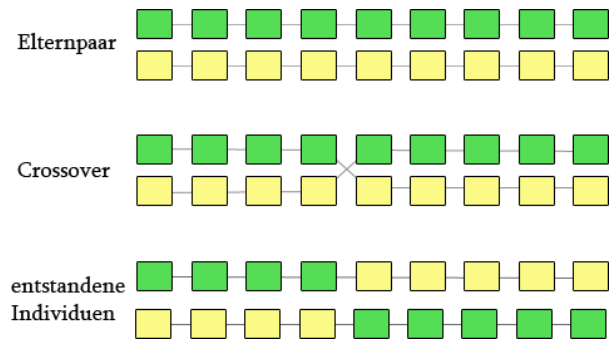


Abbildung 2.1: Crossover

Das einfachste Crossover-Verfahren ist das 1-Punkt-Crossover. Hierbei wird, wie in Abbildung 2.1 gezeigt, im Genotyp ein Punkt ausgewählt, ab dem die Gensequenzen beider Elternteile vertauscht werden [Nis97].

**Mutation** Sind alle Individuen einer Population identisch, findet statt des Crossovers eine Mutation statt. Hier werden Teile eines Individuums zufällig verändert (siehe Abbildung 2.2) [Lip06].

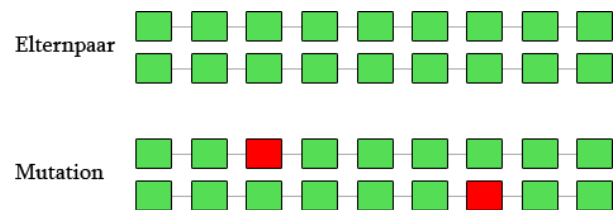


Abbildung 2.2: Mutation

**Nachkommenbewertung** Wurden Nachkommen gebildet, so werden diese anhand der Fitnessfunktion bewertet. Wenn alle Nachkommen ermittelt wurden, ist die Population vollzählig und ersetzt die vorhergehende Generation [Nis97].

## Abbruch

Es gibt mehrere mögliche Kriterien für die Terminierung des genetischen Algorithmus. Zwei der möglichen Kriterien sind ressourcenbezogen. So kann der Algorithmus beispielsweise nach einer festen Berechnungszeit enden. In Analogie dazu kann das Durchlaufen einer festen Anzahl an Populationen zum Abbruch des Verfahrens führen. [Nis97, Lip06].

Eine Alternative dazu sind qualitätsbezogene Kriterien. So kann die Berechnung abgebrochen werden, wenn die Fitness eines Individuums eine Mindestqualität erreicht. Um dieses Kriterium zu verwenden, muss es jedoch möglich sein, eine solche Mindestgüte zu definieren [Lip06].

Am Ende der Berechnung werden die besten Lösungen der Endpopulation ausgegeben [Nis97].

# Kapitel 3

## Konzeption

In diesem Kapitel soll die vorliegende Softwarestruktur näher beschrieben werden. Weiterhin sollen Ansätze aufgezeigt werden, mit deren Hilfe der Versand in einem Unternehmen automatisiert werden kann.

### 3.1 Systemarchitektur

Die im Anwendungsfall vorliegende Softwarestruktur besteht, wie in Abbildung 3.1 dargestellt, im Allgemeinen aus einem Shopsystem, sowie einem Warenwirtschaftssystem. Während das Shopsystem auf einem eigenen Server, zu dem alle Dateizugriffsrechte bestehen, liegt, ist der Zugang zum Warenwirtschaftssystem nur gemietet. Es besteht somit kein direkter Zugriff auf die zugehörigen Server, sondern lediglich auf die Weboberfläche der Anwendung. Über Im- und Exportschnittstellen können Bestellungen vom Shopsystem an das Warenwirtschaftssystem übertragen werden und die aktuellen Lagerbestände synchronisiert werden. Weiterhin können über die Oberfläche des Warenwirtschaftssystems allgemeine Exportdateien für verschiedene Versanddienstleister erstellt werden. Die Auswahl des Versanddienstes muss hierbei jedoch manuell erfolgen, da keine Automatismen implementiert sind. Weiterhin müssen trotz der Exportdateien je nach Lieferdienst noch Werte im jeweiligen weiterverarbeitenden Programm nachgetragen werden. So fehlen unter anderem die Werte für das Gewicht. Die Angabe, ob es sich um eine Nachnahmesendung handelt, ist ebenfalls nicht vorhanden.

### 3.2 Einordnung des Versandsystems

Ein automatisches Versandsystem muss Lagerdaten und auch Bestelldaten aus dem Warenwirtschaftssystem importieren. Anhand dieser Daten können die importierten Bestellungen in Pakete aufgeteilt werden. Die Paketdaten werden dann vom Programm an die Lieferdienste übermittelt. Die Paketetiketten werden anschließend einheitlich über das Versandsystem abgerufen. Eine schematische Übersicht ist in Darstellung 3.2 abgebildet. Damit eine Integration in alternative Warenwirtschafts- und Shopsysteme erleichtert wird, soll das Versandsystem objektorientiert implementiert werden. Auf diese Weise ist es auch möglich nur Teilfunktionen des Programms zu nutzen.

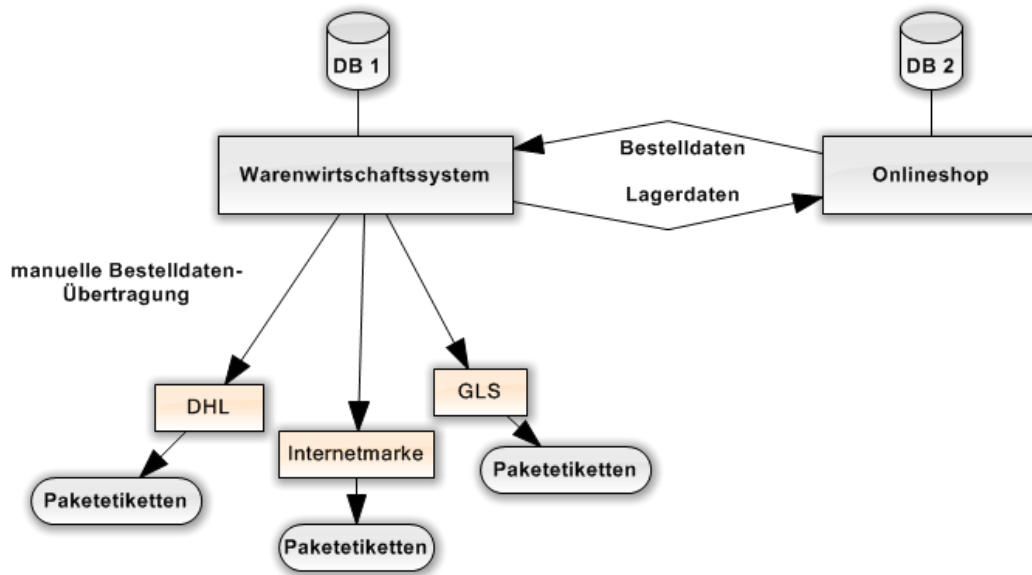


Abbildung 3.1: Vorhandene Softwarestruktur

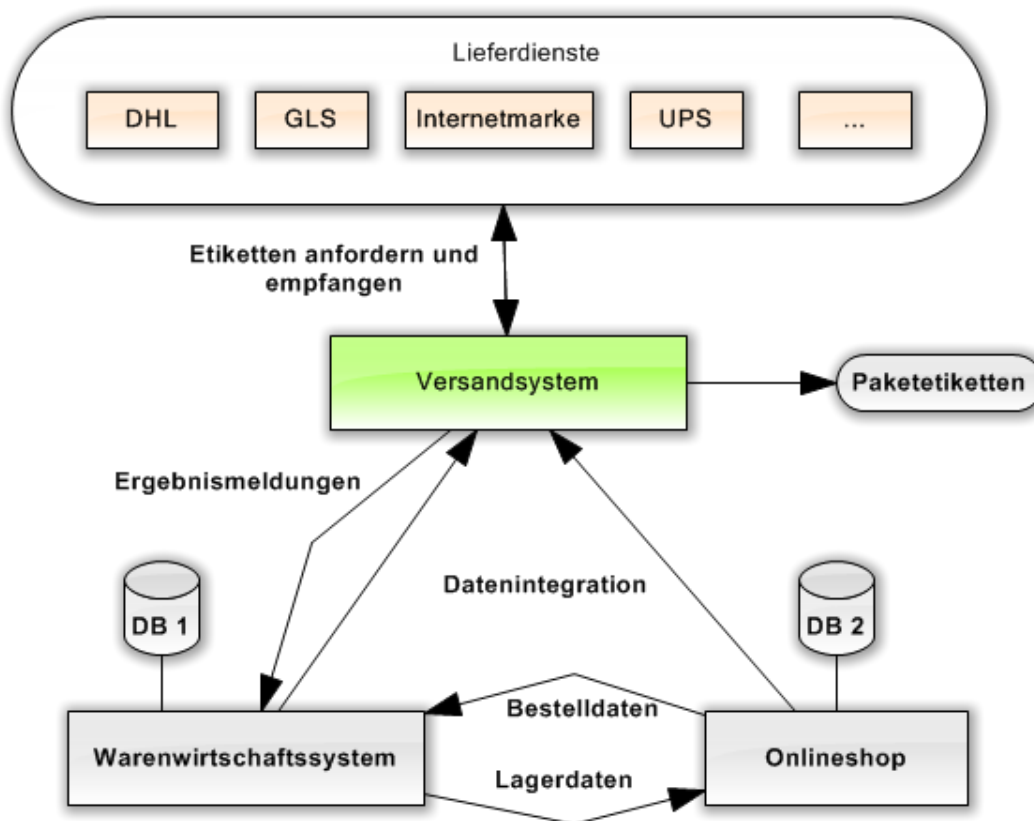


Abbildung 3.2: Einordnung

## 3.3 Datenbeschaffenheit

### 3.3.1 Bestelldaten

Die Daten der einzelnen Bestellungen liegen zentral im Warenwirtschaftssystem *DreamRobot*. Da der Zugang zum Warenwirtschaftssystem nur gemietet ist, besteht kein direkter Zugriff auf die zugehörigen Datenbanken<sup>1</sup>. Aus diesem Grund müssen die Daten auf einem anderen Weg in das Versandsystem gelangen. Dazu bieten sich im Anwendungsfall zwei Möglichkeiten an.

#### Quelltext als Datenaustauschformat

Der Benutzer kann eine Druckansicht aus *DreamRobot* kopieren und im Versandsystem einfügen. Der Quellcode dieser Ansicht wird dann vom Versandsystem analysiert und die Bestelldaten extrahiert. Über einen einheitlichen Funktionsaufruf werden die Daten nachfolgend in die internen Datenbanken des Versandsystems importiert.

Diese Methode bietet den Vorteil, dass die zu importierenden Daten von einem fachkundigen Benutzer vor dem Import angepasst werden können, sollte sich die genutzte Vorlage für die zu kopierende Druckansicht bei *DreamRobot* ändern. Der Zeitaufwand für den Benutzer ist bei dieser Vorgehensweise jedoch höher als bei der nachfolgend vorgestellten Methode, die eine CSV-Datei als Importformat nutzt.

#### Datenaustauschformat CSV

Das Warenwirtschaftssystem *DreamRobot* bietet eine Möglichkeit verschiedene Daten, wie zum Beispiel Lagerdaten, zu importieren sowie zu exportieren. Über diese Im- und Export-Schnittstelle können keine Bestelldaten exportiert werden. Als Zugriffspunkt dient eine Backupfunktion in der Bestellübersicht, die es ermöglicht, die Daten des angezeigten Monats in Form einer CSV-Datei abzurufen. Nachteilig ist, dass bei diesem Vorgang alle Bestellungen des angezeigten Monats exportiert werden. Es muss daher beachtet werden, dass alle Bestellungen, die bereits versendet wurden, gefiltert werden. Bestellungen die noch nicht ausgeliefert werden sollen, weil zum Beispiel die Zahlung nicht eingegangen ist, dürfen noch nicht im Versandsystem erscheinen.

Mit Hilfe der cURL-Bibliothek<sup>2</sup> lässt sich dieser zweite Ansatz automatisieren.

### 3.3.2 Lagerdaten

Für die im Lager befindlichen Produkte werden die für den Versand erforderlichen Daten, wie zum Beispiel Artikelnummern, Gewicht, Abmessungen und der vorhandene Lagerbestand erfasst. Sie müssen in das Versandsystem übernommen werden. Auch hierfür wird eine Schnittstelle benötigt. Im Anwendungsbeispiel befinden sich auch diese Daten in den Datenbanken des Warenwirtschaftssystems. Sie können über die

<sup>1</sup>System zur Aufbewahrung von logisch zusammenhängenden Informationseinheiten [SS].

<sup>2</sup>Kommandozeilenwerkzeug zum Hochladen, sowie zum Herunterladen von Dateien, kann auch zum Aufruf von Webseiten genutzt werden.

Im- und Export Schnittstelle direkt als CSV-Datei exportiert werden. Mit Hilfe der cURL-Bibliotheken können diese, für gewöhnlich manuellen Schritte, automatisiert werden.

Neben den herkömmlichen Lagerdaten ist es sinnvoll, die Produkte in Kategorien einzuordnen. Im Fallbeispiel werden die einzelnen Produktkategorien wie zum Beispiel “Barhocker”, “Perücken” oder “Bücher” genutzt. Diese Kategorien werden später im Verwaltungsbereich genutzt, um Einstellungen vorzunehmen, welche mehrere Produkte betreffen. Da im Fallbeispiel das Versandsystem in den Verwaltungsbereich des Shops integriert werden soll, wird angestrebt, bei der Umsetzung die Datenbank des Onlineshops zu nutzen (DB 2 in Abbildung 3.2), weil diese bereits vorhanden ist. Das hat den Vorteil, dass auf alle Daten des Shops, also auch auf die Kategoriezuordnungen, direkt zugegriffen werden kann.

### 3.4 Klassifikation von Bestellungen

Ist der Datenzugriff gewährleistet, müssen die Bestellungen in mögliche Pakete unterteilt und den Versandunternehmen zugeordnet werden, wo die Versandkosten möglichst minimal sind. Mit Hilfe von einfachen Filtern werden die Bestellungen vorsortiert. Die Lieferdienstauswahl kann anhand der vom Kunden gewünschten Zustellart, wie zum Beispiel der Lieferung per Nachnahme und anhand des Ziellandes der Bestellung, eingeschränkt werden.

Auch die Bestellung selbst kann schon vorab aufgeteilt werden. So werden Produkte ohne die erforderlichen Daten vom System ignoriert. Sie müssen später manuell nachgepflegt werden. Bereits zu diesem Zeitpunkt findet eine Einteilung der bestellten Produkte nach ihrem Lagerort statt. Zur Darstellung dieser Untergliederung werden die Produkte einer Bestellung als temporäre Pakete zusammengefasst. Diese werden in jedem Zwischenschritt weiter unterteilt.

Erst nach dem Durchlaufen der Vorverarbeitung wird jedes temporäre Teilpaket an die Funktion zur Klassifikation übergeben. Diese prüft dann anhand von Höhe, Länge, Breite und Gewicht, ob das Paket von den nach den bisherigen Schritten übrig gebliebenen Dienstleistern verschickt werden kann. Dazu sind zu jeder Versandart, die der Versanddienstleister anbietet, die entsprechenden Maße hinterlegt. Ein Auszug solcher Maße ist in Tabelle 3.1 dargestellt.

Dienstleister	Versandart	Länge	Breite	Höhe	Gewicht
DHL	Paket	120 cm	60 cm	60 cm	31,5 kg
GLS	Inlandssendung	200 cm	80 cm	60 cm	40,0 kg
Internetmarke	Großbrief	35,3 cm	25 cm	2 cm	0,5 kg
Internetmarke	Warensendung Maxi	35,3 cm	30 cm	15 cm	0,5 kg
Internetmarke	Maxibrief	35,3 cm	25 cm	5 cm	1,0 kg

Tabelle 3.1: Maximalmaße ausgewählter Lieferdienste

Besteht ein temporäres Paket aus mehreren Produkten, so muss die Produkteinteilung ermittelt werden, für die minimale Versandkosten anfallen. Auch wenn die Produkte auf Grund ihrer Größe oder ihres Gewichts von keinem Versanddienstleister in einem Paket versendet werden können, muss die Bestellung geteilt werden.

Zur Bestimmung einer Aufteilung bieten sich genetische Algorithmen oder ein Brute-Force-Ansatz an. Genetische Algorithmen haben den Nachteil, dass keine Sicherheit darüber besteht, ob das erhaltene Ergebnis auch dem Optimum der Fitnessfunktion entspricht. Weiterhin ist es sehr schwer für den speziellen Anwendungsfall geeignete Operatoren für die Mutation und den Crossover-Schritt zu finden. Je nach Abbruchkriterium kann die Laufzeit des Algorithmus bis zur Findung einer optimalen Lösung im voraus nicht bestimmt werden. Aus diesen Gründen wurde zur Klassifikation ein Brute-Force-Ansatz verwendet. Hier wird immer die optimale Lösung gefunden. Auch wenn die Berechnungszeit hier sehr hoch ist, ist diese vorher bekannt und kann im Nachhinein durch verschiedene Schritte gezielt optimiert werden.

## 3.5 Versandschnittstellen

Wenn alle Bestellungen klassifiziert wurden, müssen die Paketdaten an die Versandunternehmen übermittelt werden, um die nötigen Etiketten zu erhalten. Das Interface für die Versanddienstschnittstellen muss so aufgebaut sein, dass unterschiedliche Schnittstellen von den Versandunternehmen vom Programm einheitlich aufgerufen werden können. Weiterhin müssen neue Schnittstellen vom Programm automatisch erkannt werden.

Um eine einheitliche Kommunikation mit den verschiedenen Schnittstellen der Versanddienste zu gewährleisten, dient eine abstrakte Klasse. Von ihr erben alle im Versandsystem implementierten Schnittstellen. Die abstrakte Klasse ist in drei Teile untergliedert. Der erste Teil besteht aus den allgemeingültigen Variablendefinitionen und Methoden, die das Versandsystem für jeden Versanddienstleister benötigt. Zudem ist ein grundlegender Konstruktor hinterlegt, der die allgemeinen Daten aus der Datenbank in die Variablen der Klasse überträgt (Listing 3.1 Zeile 1 - 16 ).

Einige Lieferdienste benötigen besondere Einstellungen für die Nutzung der Schnittstelle. Hierzu zählen beispielsweise Benutzernamen, Passwörter und Links für einen automatischen Zugriff auf die Fremdsysteme. Um diese Optionen später leichter zu implementieren ist bereits ein Verwaltungssystem für vom Versanddienst abhängige Einstellungen vorbereitet (Listing 3.1 Zeile 20 - 26 ).

Um eine einheitliche Archivierung für generierte Exportdaten zu gewährleisten, ist auch eine Methode zur Generierung der Dateinamen bereits implementiert. Ein einheitlicher Dateiname macht in so fern Sinn, dass im System eine übersichtliche Verzeichnisstruktur für das spätere Auffinden von bereits gedruckten Versandetiketten herrscht (Listing 3.1 Zeile 28 - 30 ). Alle anderen Methoden sind von Lieferdienst zu Lieferdienst zu unterschiedlich, um auf diese Weise vorbereitet zu werden.

Listing 3.1: Allgemeingültige Variablen und Methoden

```
1 protected $id;  
2 protected $dispatcher;  
3 protected $dispatchType;
```

```

4  protected $price;
5  protected $maxMeasures;
6  protected $allows_COD;
7  protected $allowedCountries;
8  protected $option;
9  protected $color;
10 protected $storno;
11 protected $islandsurcharge;
12 protected $isbulk;
13
14 public function __construct($id){
15     Hole Lieferdienst mir der ID $id aus Datenbank;
16 }
17
18 /* Getter und Setter für obige Variablen, der Übersicht halber weggelassen */
19
20 protected final function setSetting($key, $value, $id='') {
21     Setze Einstellungen für Schlüssel $key auf Wert $value bei Lieferdienst mit ID $id;
22 }
23
24 protected final function removeSetting($key, $id='') {
25     Entferne Einstellungen für Schlüssel $key bei Lieferdienst mit ID $id;
26 }
27
28 protected final function getFileName($filename, $extension, $content=array()){
29     Gebe einen Pfad für die Speicherung von Datei $filename mit der Dateinamenserweiterung
30     $extension zurück;
31 }

```

Im zweiten Teil der Klasse kommen Methodendefinitionen dazu, die zum Druck von Versandetiketten notwendig sind. Da mehrere Pakete, die an die gleiche Adresse gesendet werden, von den Lieferdiensten oft als ein Auftrag zusammengefasst werden, sammeln die Schnittstellen im Versandsystem mit der `appendData()`-Methode zunächst alle Versandaufträge für die sie zuständig sind (Listing 3.2 Zeile 1). Erst der Aufruf der `submit()`-Methode übermittelt die Daten gebündelt an das Versandunternehmen (Listing 3.2 Zeile 3). Eine dritte Methode, die in diese Kategorie zählt, ist `pseudoSubmit()` (Listing 3.2 Zeile 5). Diese generiert alte Exportdaten neu, ohne jedoch eine Bestellung an den Versanddienstleister abzusenden. Dies ist gewünscht, um verlorene Etiketten ein weiteres Mal drucken zu können.

Listing 3.2: Pflichtangaben

```

1  abstract function appendData($billnr, $packid);
2
3  abstract function submit();
4
5  abstract function pseudoSubmit();

```

Im dritten Abschnitt der abstrakten Klasse sind in Listing 3.3 Methoden angegeben, die bereits implementiert sind. Hierzu zählt die Methode zur Preisberechnung, Methoden zum Stornieren von Etiketten sowie Methoden, um die Optionsseiten des Lieferdienstes zu erweitern.

Während die Methode zur Preisberechnung (Zeile 1 -14) auf Grund der variierenden Berechnung zwischen den Versanddienstleistern meist überschrieben werden muss, trifft dies auf die anderen Methoden nicht zu. Da nicht alle Versanddienstleister eine Stornierung ihrer Etiketten anbieten, muss auf die Methoden `appendStorno()` und `getStorno()` teilweise verzichtet werden. Eine Implementierung dieser Funktionen wird vom System auch nicht vorausgesetzt. Die `appendStorno()`-Methode wird wie `appendData()` aus Listing 3.2 genutzt um Pakete für die Stornierung vorzumerken. Die Methode `getStorno()` baut dann eine Ansicht zusammen, über die der Benutzer seine Eingaben nochmals prüfen

kann (Zeile 16-22). Um besondere Optionsseiten implementieren zu können, dienen die letzten drei Methoden in Listing 3.3. `listOptionPanes()` gibt einen Array zurück, mit dessen Hilfe neue Menüeinträge vom System angelegt werden. `showOptionPanes()` zeigt die Einstellungsseiten dann im System an, während `calcOptionPanes()` die Verarbeitung von eingegebenen Daten übernimmt. Da einige Versandsdienste auch mit den Grundeinstellungen, die das Versandsystem bereits bietet, auskommen, ist es nicht immer erforderlich diese Methoden zu implementieren.

Listing 3.3: Erweiterbare Methoden

```
1 public function getPrice($param){
2     return $this->price;
3 }
4
5 public function appendStorno($packnr){
6     return false;
7 }
8
9 public function getStorno(){
10    return false;
11 }
12
13 public function listOptionPanes(){
14    return array();
15 }
16
17 public function showOptionPanes($formhead){
18    return '';
19 }
20
21 public function calcOptionPanes()    {
22 }
```

Zu diesen Methoden kommen noch Methoden zum Installieren und Deinstallieren einer neuen Schnittstelle. In diesen Methoden können je nach Bedarf Änderungen an der Datenbank durchgeführt oder neue Einstellungsmöglichkeiten im System eingebunden werden.



# Kapitel 4

## Implementierung

In diesem Kapitel wird auf die Implementierung, sowie konkretere Programmabläufe des im Anwendungsfall implementierten Versandsystems eingegangen.

### 4.1 Klassifikation von Paketen

Als eine der umfangreichsten Kernfunktionen hat sich die automatische Klassifikation von Paketzusammenstellungen herausgestellt. Bevor die Artikel anhand ihrer Abmessungen in möglichst günstige Pakete unterteilt werden, durchläuft die Bestellung eine Reihe von Funktionen. Hierbei werden die Bestellungen nach verschiedenen Kriterien aufgeteilt (Vorverarbeitung) und schließlich an die eigentliche Erkennung übergeben (siehe Abbildung 4.1).

#### 4.1.1 Vorverarbeitung

Vor Beginn der Klassifikation durchlaufen die Bestellungen zunächst verschiedene Verarbeitungsschritte, um die spätere Erkennung zu vereinfachen.

##### **Versanddienstfilter** (`initCheck()`)

So werden bei jeder Bestellung zunächst die in der Datenbank hinterlegten Lieferdienste gefiltert.

Einige Versandunternehmen liefern nur innerhalb Deutschlands, so dass diese bei Bestellungen, die ins Ausland geliefert werden sollen, nicht weiter berücksichtigt werden müssen. Weiterhin wird auch nicht für alle Versandarten die Möglichkeit angeboten Pakete per Nachnahme zu versenden. Auch diese Versandunternehmen werden in diesem Schritt aussortiert. In der Tabelle 4.1 ist ein Auszug aus der Konfiguration im Anwendungsfall dargestellt.

Durch die hinterlegten Filter entfallen verschiedene Versandmöglichkeiten. Unter bestimmten Umständen bleibt dann nur noch eine Möglichkeit, um das Paket zu versenden. Das kann die Berechnung stark abkürzen. Ein Beispiel hierfür wäre, bei

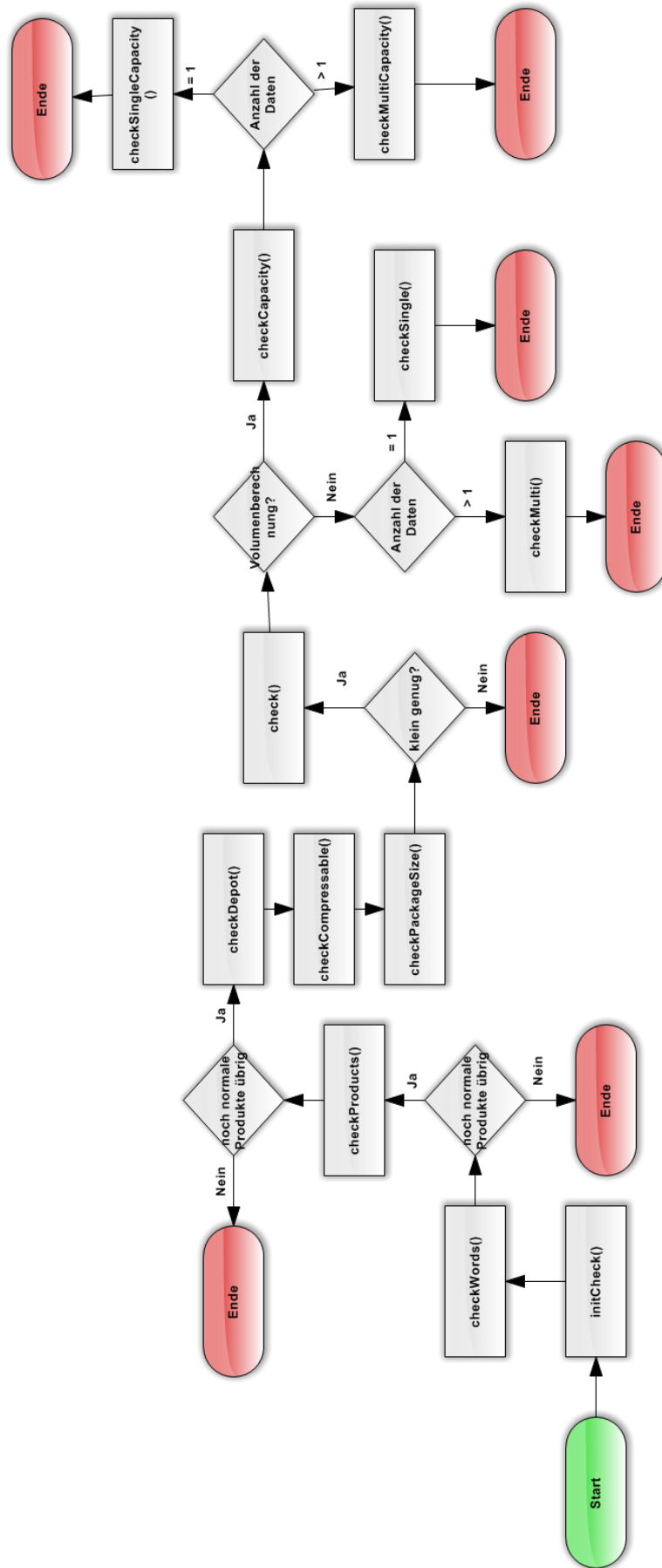


Abbildung 4.1: Ablauf der Klassifikation

Dienstleister	Lieferart	erlaubte Länder	Nachnahme
DHL	Paket	DE	Ja
DHL	Sperrgut	DE	Ja
GLS	National	DE	Ja
GLS	Europa	GB,CH,AT,LU,NL,FR	Ja
Internetmarke	Großbrief	DE	Nein
Internetmarke	Warensendung Maxi	DE	Nein
Internetmarke	Maxibrief	DE	Nein
Internetmarke	Großbrief - International	GB,CH,AT,LU,NL,FR	Nein

Tabelle 4.1: Filterkriterien

Nutzung der in Tabelle 4.1 dargestellten Konfiguration, eine Sendung nach Frankreich, die per Nachnahme verschickt werden soll. In diesem Fall bleibt nur die Lieferart “Europa” von GLS übrig. Es muss in den nachfolgenden Verarbeitungsschritten nur noch eine Aufteilung in Pakete gefunden werden, die den gültigen Maßen von GLS entsprechen.

Die nötigen Daten bekommen die Funktionen aus der Bestellung. Zielland und auch Zahlungsart wurden beim Datenimport aus dem Warenwirtschaftssystem übernommen.

### **Lernsystem (initCheck())**

Als eine Funktion, die wahlweise benutzt werden kann, zählt das Lernsystem, welches Bestellungen anhand von bereits gesammelten Daten einteilt. Hierbei wird die aktuelle Bestellung mit vergangenen Bestellungen verglichen. Sofern die Rahmenbedingungen wie Zielland, Nachnahmesendung und auch die Paketzusammensetzung übereinstimmen, wird die Art der Lieferung von dem Paket aus der Datenbank übernommen. Um die Arbeitszeit, die das Lernsystem zum Suchen von ähnlichen Bestellungen verwendet zu verkürzen, werden an dieser Stelle nicht die Abmessungen der vergangenen Bestellungen geprüft. Stattdessen wird eine Übereinstimmung der enthaltenen Artikelnummern gesucht. Das dieses Kriterium genügt, wird sich später im Rahmen der Evaluation zeigen.

Je nach Konfiguration des Systems werden an dieser Stelle entweder nur manuell veränderte Paketzusammenstellungen oder alle bislang erkannten Bestellungen betrachtet. Bei erfolgreicher Erkennung durch das Lernsystem durchläuft die Bestellung nicht mehr die komplette Klassifikation. Der Erkennungsvorgang wird dadurch beschleunigt.

### **Artikelfilter (checkWords())**

Bestellungen können auch Positionen enthalten, bei denen es sich nicht um Produkte handelt. Hierbei handelt es sich im konkreten Anwendungsfall um Rechnungspositionen für besondere Zahlarten, wie zum Beispiel PayPal oder der Lieferung per Nachnahme, die als Produkte im Warenwirtschaftssystem hinterlegt sind. Auch Gutschriften, Rabatte sowie Retouren werden an dieser Stelle im System erfasst. Solche Positionen werden aus der Bestellung herausgefiltert. Da die Bezeichnungen für diese Positionen sehr einheitlich sind, besteht im Verwaltungsbereich des Versandsystems die Möglichkeit, einen

Wortfilter anzulegen, der dafür sorgt, dass diese bei der Berechnung der Paketaufteilung ignoriert werden.

### Datenvollständigkeit (`checkProducts()`)

Nach diesem Schritt wird geprüft, ob für alle Produkte, die in der Bestellung enthalten sind, auch die erforderlichen Daten wie Abmessungen, Gewicht und Lagerort vorliegen und ob genügend Artikel dieser Art vorrätig sind.

Nur bei Vollständigkeit der Angaben kann die Berechnung fortgesetzt werden. Andernfalls werden die entsprechenden Artikel separat zurückgegeben und mit einer Fehlermeldung versehen. Diese gibt dem Benutzer Auskunft darüber, welche Daten fehlen. Die übrigen Produkte werden vom System weiter berechnet.

### Vorsortierung (`checkDepot()`)

Verbleibende Produkte werden im nächsten Schritt nach ihrem Lagerort sortiert. Aus Gründen der Wirtschaftlichkeit wird versucht alle Produkte von einem Lagerort aus zu versenden. Es kommt jedoch vor, dass Artikel an verschiedenen Orten lagern. Tritt dieser Fall ein, so muss die Bestellung geteilt werden.

Enthält die Bestellung zudem noch Artikel, die an mehreren Lagerorten auf Vorrat sind, so werden alle folgenden Schritte für jede Zuordnungsmöglichkeit dieser Produkte berechnet (Brute-Force-Ansatz).

Das Beispiel in Abbildung 4.2 soll verdeutlichen, welche Möglichkeiten sich in einem solchen Fall ergeben können. Hier geht eine Bestellung ein, bei der nicht alle Produkte an einem Lagerort vorliegen. Die Produkte 1 und 3 befinden sich jeweils nur in einem Lager. Eine Aufteilung des Versands ist damit unausweichlich. Die Produkte 2 und 4 liegen an beiden Lagerorten vor. Es muss also ermittelt werden, welche der vier in Abbildung 4.2 aufgelisteten Teilungsmöglichkeiten günstiger ist. Es werden daher alle vier Kombinationen geprüft.

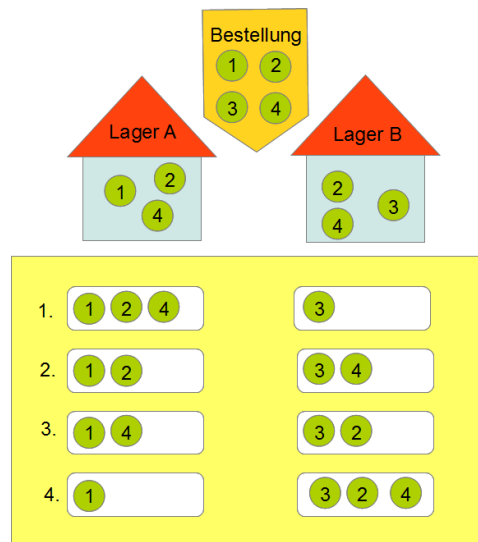


Abbildung 4.2: Lageraufteilungsmöglichkeiten

### Produktintegration (`checkCompressable()`)

Besteht eine Bestellung aus mehreren Produkten, so wird im nächsten Schritt geprüft, ob kleinere Produkte in größere gepackt werden können. Dies ist beispielsweise der Fall, wenn im Karton eines Artikels noch Platz für andere Produkte ist. So kann man zum Beispiel eine Perücke, die in einer elastischen Tüte verpackt ist, in den Karton eines Motorradhelms mit hineinpacken. Die Abmessungen des kleineren Produkts werden bei der

nachfolgenden Erkennung nicht weiter beachtet. Einzig das Gewicht wird bei der weiteren Berechnung beachtet. Die Zuordnung, welche Produkte in welche anderen Produkte gepackt werden können, lässt sich im Verwaltungsbereich des Versandsystems verändern.

### 4.1.2 Erkennung

An dieser Stelle ist die Vorverarbeitung abgeschlossen und das System hat bereits erste temporäre Pakete erstellt. Diese Pakete durchlaufen nun die eigentliche Klassifikation, bei der sie entweder noch weiter unterteilt werden oder aber den hinterlegten Lieferdiensten direkt zugeordnet werden.

Die Zuordnung erfolgt im Allgemeinen über die Abmessungen der einzelnen Produkte. Es gibt jedoch auch Ausnahmen, bei denen die Erkennung über die Abmessungen nicht direkt anwendbar ist. Dies betrifft Produkte wie Flüssigkeiten, Perücken oder Kleidungsstücke. Durch die leichte Verformbarkeit solcher Artikel ist es möglich, eine Seitenlänge zu Lasten einer anderen zu verringern, indem man beispielsweise ein T-Shirt in der Mitte faltet. Das kann dazu führen, dass andere Versandarten möglich werden. Bei der Datenpflege werden diese Produkte, wie alle anderen auch, ausgemessen. Ob ein Produkt anstatt über die Abmessungen über das Volumen klassifiziert werden soll, muss im Verwaltungsbereich eingestellt werden.

#### Volumenklassifikation

Wenn alle Produkte in einem temporären Paket über das Volumen klassifiziert werden können, werden diese an eine entsprechende Funktion übergeben. Hier wird aus den ausgemessenen Werten des Produkts das Volumen berechnet und mit dem maximalen Volumen verglichen, welches in die jeweilige Versandart passt.

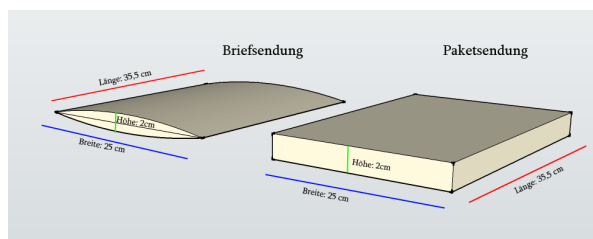


Abbildung 4.3: Gegenüberstellung Brief- und Paketsendung

Bei Briefsendungen kann diese Berechnung auf Grund von deren Form zu Fehlern führen. So hat ein Großbrief die maximalen Abmessungen von 35,3 cm Länge, 25 cm Breite und 2 cm Höhe ( $= 1765\text{cm}^3$ ). Doch sind Briefe nicht kubisch und haben somit kein Volumen von  $1765\text{cm}^3$  (siehe Abbildung 4.3). Das reale Volumen liegt etwa bei der Hälfte, variiert jedoch, je nach verwendeter Umschlagsform. Daher müsste theoretisch je nach Form des Umschlags, beziehungsweise Pakets, die Volumenberechnung über eine andere Formel erfolgen. Um dieses Problem zu vereinfachen, bekommt der Benutzer nur die Möglichkeit einen Pufferwert einzustellen, der bei der Volumenberechnung mit einkalkuliert wird.

Zur Erkennung werden alle möglichen Produktkombinationen gebildet. Dazu werden zunächst die zu klassifizierenden Produkte durchnummeriert. Von den Nummern werden

durch mehrfaches Aufrufen der Methode `nextPermutation()` aus dem Listing 4.1 alle möglichen Permutationen gebildet. Aus jeder Permutationsfolge werden anschließend Produktkombinationen zusammengefügt.

Hierzu werden in der Methode `generateCombinations()` schrittweise die Nummern in einer Permutationsfolge zusammengefasst. Jeder der in der Methode gebildeten Zwischenschritte beinhaltet bereits eine gültige Kombination (siehe Zeile 43). Zuletzt werden die Nummern wieder den Produkten zugeordnet, und somit potentielle Paketkombinationen gebildet. Es erfolgt ein Vergleich des Volumens aller Pakete in einer solchen Kombination mit dem Volumen, welches in die maximalen Abmessungen der einzelnen Versandarten passt. Von allen Paketkombinationen, die eine vollständige Zuordnung zu den Versandarten ermöglichen, wird dann die kostengünstigste ausgewählt.

Listing 4.1: Auszug: Bildung von Paketkombinationen

```
1  function nextPermutation($n)
2  {
3      $add = ($this->intAnzahl - 1);
4      // Restplätze durchlaufen
5      for ($i = $n; $i < $this->intAnzahl; $i++)
6      {
7          // Tauschen der aktuellen Position mit der übergebenen Position
8          $intHelp = $this->arrZahlen[$i];
9          $this->arrZahlen[$i] = $this->arrZahlen[$n];
10         $this->arrZahlen[$n] = $intHelp;
11
12         // Falls das Ende der Zahlenmenge noch nicht erreicht wurde
13         if ($n < ($this->intAnzahl - 1))
14         {
15             $this->nextPermutation($n + $add);
16         } else
17         {
18             $this->permutations[] = array_reverse($this->arrZahlen);
19         }
20
21         // Tauschen wieder rückgängig machen
22         $intHelp = $this->arrZahlen[$i];
23         $this->arrZahlen[$i] = $this->arrZahlen[$n];
24         $this->arrZahlen[$n] = $intHelp;
25     }
26 }
27
28 function generateCombinations($perm)
29 {
30     $result = array();
31     $a = array();
32     $c = count($perm);
33     for ($i = 0; $i < ($c - 1); $i++)
34     {
35
36         $a = array_merge($a, array(array_shift($perm)));
37
38         if (count($a) > 1)
39         {
40             $b = $this->generateCombinations($a);
41             foreach ($b as $d)
42             {
43                 $result[] = array_merge($d, array($perm));
44             }
45         }
46
47         $result[] = array($a, $perm);
48     }
49     return $result;
50 }
```

## Klassifikation über die Abmessungen

Bei einer Vielzahl an Produkten ergibt es keinen Sinn eine Berechnung über das Volumen durchzuführen. Als Beispiel hierfür lassen sich Motorradhelme nennen. Diese sind nicht verformbar und befinden sich bereits in einem Karton mit festen Abmessungen. In diesem Fall nutzt das System die herkömmliche Klassifikation.

Abweichend zur Klassifikation über das Volumen muss hier für jede Packkombination ermittelt werden, wie Pakete, die aus mehreren Produkten bestehen, zusammengesetzt werden. Dazu werden die Produkte für jedes Paket der Größe nach sortiert. Für den Vergleichswert, nach dem sortiert wird, kommen mehrere Angaben in Frage. Dazu zählen unter anderem das Volumen des Produkts, die Summe der Kantenlängen und die längste Seite des Produkts. Da sich in einem empirischen Versuch herausstellte, dass die verschiedenen Sortierkriterien alle zu den gleichen Ergebnissen führten, erfolgt die Sortierung anhand der längsten Kante der Produkte, da hier keine zusätzlichen Rechenoperationen erforderlich sind.

Die jeweils kleinsten Produkte werden dann so kombiniert, dass der umschließende Quader ein minimales Volumen erhält. Anschließend wird der erhaltene Quader mit dem nächsten Produkt auf die gleiche Weise kombiniert. Dieser Vorgang wird so lange wiederholt, bis alle Produkte im Quader enthalten sind. Die Abmessungen des Quaders entsprechen dann denen des potentiellen Pakets.

## Sonderfälle

Es gibt einige Sonderfälle, bei denen die Klassifikation weiter beschleunigt werden kann. Diese können eintreten, wenn alle Produkte der Bestellung identisch ist, die Bestellung aus zu vielen Produkten besteht oder Gefahren- oder Luxusgüter enthält.

**Alle Produkte sind identisch.** Besteht eine Bestellung ausschließlich aus dem gleichen Produkt, so werden nicht alle möglichen Permutationen und deren Kombinationen gebildet, sondern lediglich die Kombinationen der Ausgangsmenge der Permutationen, da die restlichen Kombinationen in dem Fall identisch sind.

**Die Bestellung besteht aus mehr als x Produkten.** Je nach Rechenkapazität des Servers, auf dem das Versandsystem läuft, kann die Berechnungsgeschwindigkeit variieren. Da die Rechendauer bei dem verwendeten Verfahren exponentiell mit der Anzahl der Produkte steigt, kann die Berechnung teilweise sehr lange dauern. Um dem Benutzer unnötige Wartezeit zu ersparen, kann die Erkennung ab einer frei konfigurierbaren Anzahl an Produkten abgebrochen werden. In diesem Fall ist eine manuelle Klassifizierung durch den Anwender effizienter. Entsprechende Bestellungen werden durch einen Hinweis kenntlich gemacht.

**Die Bestellung enthält Gefahren- oder Luxusgüter.** Bei Gefahrgütern müssen verschiedene Richtlinien für den Transport beachtet werden. Auch teurere Produkte sollen in einer dem Preis angemessenen Verpackung versendet werden. Für diesen Fall

werden in der Verwaltung Sonderregeln erstellt. So kann angegeben werden, ob einzelne Dienstleister für die Versendung von bestimmten Produkten ausgeschlossen werden sollen. Diese Sonderregeln werden vor der Klassifikation eines Paketes geprüft.

## 4.2 Benutzeroberfläche

Wenn alle Bestellungen die Klassifikation durchlaufen haben, erhält der Benutzer eine Übersicht, in der die automatisch erkannten Werte noch einmal überprüft werden können.

**Versandautomatisierung - Hauptseite (45)**

Hauptseite (43) | **Bestelldaten importieren (0)** | Letzte Exporte | Verlauf (0) | Suchen | ausgeblendete Bestellungen (21) | Löschliste (0) | Stornieren | Versandkostenübersicht | Verwaltung |

• **Achtung, es liegt mindestens eine Bestellung mit einem Problem vor.**

Exportdaten generieren | Bearbeitungsmodus | Daten neu berechnen | Bestellungen ausblenden | Bestellungen löschen | Lagerdaten neuladen

WarenNr.	Kundenname	Infos	Land	<input checked="" type="checkbox"/>	Pakete	Versandort	Eigentümer	Versanddienst
27708	Frankfurt		DE	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 2 x weibliche Schaufensterpuppe schwarz-glänzend faceless (425)</li> <li>• 1 x Nachnahmegebühren (0)</li> </ul>	Gransee	Christian	DHL - Paket
27707	Frankfurt		DE	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 1 x Grau-Weiße Perücke (825)</li> <li>• 1 x Paypalgebühren (0)</li> </ul>	Gransee	Christian	Internetmarke - Warensendung Maxi
27705	Frankfurt		DE	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 1 x EASTPAK Rucksack WALKER Grey Pinstripe 49x34x24 cm EK207 (0)</li> </ul>	Gransee		
27695	Frankfurt		DE	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 1 x RL 2x Barhocker weinrot ECHT LEDER Clemens CQ-220wineredrl_2Stück (1295)</li> </ul>	Gransee	Tanyel + Christian	DHL - Paket
				<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 1 x RL 2x Barhocker weinrot ECHT LEDER Clemens CQ-220wineredrl_2Stück (1295)</li> </ul>	Gransee	Tanyel + Christian	DHL - Paket
				<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 1 x kostenloser Versand (0)</li> </ul>	Gransee		
27694	Frankfurt		DE	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 1 x Bistro Tisch weiß rund Holz Platte TC3-1036B-3white (1973)</li> </ul>	Gransee	Tanyel + Christian	GLS - National
				<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 1 x kostenloser Versand (0)</li> </ul>	Gransee		
27692	Frankfurt		DE	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>• 2 x Gelbe Perücke (1599)</li> </ul>	Gransee	Christian	Internetmarke - Grossbrief

Abbildung 4.4: Versandvorschläge

Durch farbliche Kennzeichnung kann der Benutzer feststellen, ob bei der Erkennung Fehler aufgetreten sind. Er erhält darüber hinaus Auskunft über die Details der zusammengesetzten Pakete, wie den Lagerort und den Versanddienstleister.

Fehler werden durch ein Ausrufezeichen gekennzeichnet und standardmäßig in Rottönen hervorgehoben. Rot, als eine der auffälligsten Farben, gilt im Kontrast mit Weiß oder Schwarz als Warnfarbe [Sch10]. Beim Mouse-over des Ausrufezeichens, welches sich im Infobereich einer Bestellung befindet, erscheint ein Tooltip, der den Benutzer über die Art der aufgetretenen Fehler informiert. Hierbei wird in kritische und unkritische Fehler unterschieden.

Fehler zählen als kritisch, wenn die Klassifikation nicht erfolgreich beendet wurde und somit keine Zuordnung zu einem Lieferdienst stattfand. Solche Bestellungen können ohne eine manuelle Korrektur nicht versendet werden. Der kritische Fehler in Abbildung 4.4 ergibt sich beispielsweise aus einer fehlenden Zuordnung. Da die Artikelnummer eines



Produkts nicht angegeben wurde, konnten keine Daten ermittelt werden. Der Erkennungsvorgang wurde frühzeitig abgebrochen.

Bei den unkritischen Fehlern handelt es sich um Fehler bei der Datenpflege. Das Programm erkennt, dass einzelne Abmessungen fehlen. Da die Software nicht feststellen kann, ob dies beabsichtigt ist, gibt sie nur den Warnhinweis über die fehlenden Werte aus, da dadurch die Berechnung der betreffenden Bestellung falsch sein könnte.

Wie in der Abbildung 4.4 zu erkennen ist, wird jedem Versanddienst eine Farbe zur besseren Übersicht zugeordnet. Damit sich die Fehlerfarben auch bei Hinzukommen neuer Versanddienste noch von deren Farben abheben, ist es möglich, den Fehlertypen über die Programmeinstellungen andere Farben zur Hervorhebung zuzuweisen. In der Abbildung 4.4 wird für die kritischen Fehler ein helles Rot und für die unkritischen Fehler Violett verwendet, da sich diese Farben von allen anderen im System konfigurierten gut abheben.

Einige Kunden hinterlegen beim Bestellvorgang Kommentare, die auf besondere Liefergegebenheiten, wie beispielsweise eine verzögerte Lieferung, hinweisen. Um diese Informationen schnell abrufen zu können, erscheint ein Tooltipp beim Mouse-over des Informationszeichens. Da diese Informationen im laufenden Betrieb leicht zu übersehen sind, gibt es die Möglichkeit, einen zusätzlichen Warnhinweis zu aktivieren. Dieser Warnhinweis dient jedoch nur zur schnelleren Übersicht und erfordert kein zwingendes Eingreifen vom Benutzer.

Das Eurosymbol im Infobereich liefert, bei Mouse-over, Informationen über die berechneten Versandkosten zur jeweiligen Bestellung. Dies dient hauptsächlich der Kontrollmöglichkeit durch den Benutzer.

Sind eventuell auftretende, kritische Fehler behoben, so können unter dem Menüpunkt "Exportdaten generieren" alle gelisteten Bestellungen über die Schnittstellen zu den jeweiligen Lieferdiensten übertragen werden. Einige solcher Schnittstellen sind als einzelne Module bereits implementiert. Zur Erweiterung der Software um neue Versandunternehmen ist es nötig, neue Module zu entwickeln. Die einzelnen Schnittstellen können in ihrer Funktionsweise variieren. Einige liefern lediglich eine CSV-Datei zurück, die dann manuell auf der Webseite des Lieferdienstes eingelesen werden muss. Andere wiederum liefern direkt PDF-Dateien mit den entsprechenden Paketetiketten zurück. Alle Dateien sind nach den Versandorten der Pakete unterteilt, sodass der Benutzer diese nur noch auf dem jeweiligen Drucker vor Ort ausfertigen muss.

### 4.3 Integrierbarkeit

Da das Versandsystem auch als alleinstehendes System arbeiten kann, sind für eine Integration in beliebige Shopsysteme nur drei Zeilen Code erforderlich. Einzig die Importschnittstellen müssen, sofern sie für das entsprechende Shopsystem nicht bereits bestehen, zusätzlich noch implementiert werden.

Listing 4.2: Befehlsaufruf zur Integration

```
1 define("AS_FOLDER", 'automated_shipping/');  
2 include_once (AS_FOLDER . '/Main.php');
```

```
3 Main::show();
```

So wird in Zeile 1 des Listings 4.2 der Verzeichnispfad zu den Quelldateien des Versandsystems definiert. In Zeile 2 wird die Hauptdatei eingebunden und in der dritten Zeile wird eine Methode zur Anzeige aufgerufen. Wahlweise können noch CSS-Dateien eingebunden werden, um die Anzeige an die eigenen Bedürfnisse anzupassen.

Das xt:commerce Shopsystem im Fallbeispiel verfügt bereits über eine eigene Session<sup>1</sup>-Verwaltung. Um Konflikte zu vermeiden, wurde beim Versandsystem gänzlich auf die Verwendung von Sessions verzichtet. Sämtliche Übergaben von Variablen werden über GET- und POST-Methoden realisiert. Der Nachteil, den dies mit sich bringt, soll jedoch nicht verschwiegen werden. Da die Datenmengen, die im System übergeben werden, an einigen Stellen sehr groß sind, kann dies die Serverbeschränkungen schnell überschreiten. In diesem Fall müssen, damit das System fehlerfrei arbeiten kann, die Serverbeschränkungen angehoben werden. Wie hoch die Datenmengen bei den Übergaben sind, hängt stark von der Anzahl der zu verarbeitenden Bestellungen ab.

Aus Kompatibilitätsgründen wurde auch auf ein eigenes Rechtesystem verzichtet, da jedes Shop-System bereits über umfangreiche Sicherheitssysteme verfügt. Ein direkter Zugriff auf das Versandsystem ist durch die objektorientierte Programmierweise nicht möglich. Erst durch das Einbinden des Quellcodes aus Listing 4.2 kann ein Zugriff erfolgen. So wird das Versandsystem im Fallbeispiel durch eine Zugriffssteuerung des Shopsystems geschützt, da hier bei der Integration des Versandsystems auch eine Rechteprüfung vom Shopsystem eingebunden wird. Das Nachrüsten eines eigenen Rechtesystems ist jedoch problemlos über Module möglich. Dies bietet sich an, wenn das Versandsystem als Stand-Alone-Anwendung verwendet werden soll.

Eine weitere Integrationsmöglichkeit bietet eine API, mit der die Versandkosten vor dem Auslösen der Bestellung berechnet werden können. Mit Hilfe dieser API ist es möglich, dem Kunden bei der Bestellung genau den Versandpreis zu berechnen, der für den Händler anfällt.

---

<sup>1</sup>Durch eine ID bestimmte Instanz bei Webanwendungen.

# Kapitel 5

## Evaluation

In diesem Kapitel soll nun die entwickelte Software getestet werden. So soll hier gezeigt werden, wie effizient die Klassifikation auf Basis des Brute-Force-Algorithmus arbeitet und auch wie leistungsstark die einfachen Vergleichsoperationen unter Verwendung des Lernsystems ausfallen.

### 5.1 Berechnungszeit

Das wohl auffälligste Problem an der Implementierung ist die Nutzung eines Brute-Force-Ansatzes zur Findung von Paketen und der damit verbundene Zeitaufwand. Die Tabelle 5.1 zeigt die Berechnungszeit für unterschiedliche Bestellungen. Alle Tests wurden auf zwei Rechnern durchgeführt. Bei *Rechner A* handelt es sich um einen Intel Pentium 4 mit 3,00 GHz und 2GB RAM. *Rechner B* ist mit einem DualCore Prozessor mit 2 x 2,8 GHz und 4 GB RAM etwas leistungsstärker, was sich stark auf die Rechenzeiten auswirkt.

In jedem Testlauf wurde eine identische Bestellung fünfzig mal durchlaufen, um Belastungsschwankungen der Rechner auszugleichen. Jede der berechneten Bestellgrößen wurde einmal ausschließlich mit Produkten, die vom System über das Volumen klassifiziert werden und einmal mit Produkten, die über die Maße erkannt werden, getestet. Zum Vergleich sind am Ende der Tabelle Messungen aus dem realen Betrieb, mit gemischten Daten, angegeben. Bei allen Messungen war das Lernsystem deaktiviert, um die reine Berechnungszeit zu bestimmen.

Der Tabelle ist zu entnehmen, dass die durchschnittliche Rechenzeit mit Zunahme der Produkte in einer Bestellung exponentiell steigt. Der Grund für diesen Anstieg liegt bei der Nutzung der Brute-Force-Methode. So müssen mit Zunahme der Produkte mehr Kombinationen geprüft werden. Die Anzahl der Kombinationen entspricht hierbei  $A_K = n! \cdot \sum_{i=0}^n \frac{(-1)^i}{i!} \cdot n \cdot l$  wobei  $n$  der Anzahl der Produkte entspricht und  $l$  der Anzahl der Lieferdienste.

Die durchschnittliche Rechenzeit bei Bestellungen mit mehr als 8 Produkten ist sowohl bei der Berechnung über das Volumen, als auch bei der Berechnung über die Abmessungen, sehr hoch. Da der Versand nahezu in Echtzeit berechnet werden soll, werden im Versandsystem Bestellungen, die aus zu vielen Produkten bestehen, bei der Berechnung übersprungen, um dem Benutzer lange Wartezeiten zu ersparen. Dies betrifft in einem

	Gesamtlaufzeit - Rechner A	$\emptyset$ -Laufzeit - Rechner A	Gesamtlaufzeit - Rechner B	$\emptyset$ -Laufzeit - Rechner B
Bestellungen aus 1 Produkt (Volumen)	22 s	0,44 s	11 s	0,22 s
Bestellungen aus 3 verschiedenen Produkten (Volumen)	31 s	0,62 s	14 s	0,28 s
Bestellungen aus 6 verschiedenen Produkten (Volumen)	351 s	7,02 s	141s	2,82 s
Bestellungen aus 9 verschiedenen Produkten (Volumen)	6699 s	133,98 s	1586 s	31,72 s
Bestellungen aus 9 gleichen Produkten (Volumen)	647 s	12,94 s	204 s	4,08 s
Bestellungen aus 1 Produkt (Maße)	22 s	0,44 s	11 s	0,22 s
Bestellungen aus 3 verschiedenen Produkten (Maße)	36 s	0,72 s	15 s	0,3 s
Bestellungen aus 6 verschiedenen Produkten (Maße)	411 s	8,22 s	141 s	2,82 s
Bestellungen aus 9 verschiedenen Produkten (Maße)	7357 s	147,14 s	2198 s	43,96 s
Bestellungen aus 9 gleichen Produkten (Maße)	809 s	16,18 s	231 s	4,62 s
reale (gemischte) Bestellungen (1)	28 s	0,56 s	15 s	0,3 s
reale (gemischte) Bestellungen (2)	37 s	0,74 s	13 s	0,26 s

Tabelle 5.1: Berechnungszeiten für je 50 Bestellungen

Test 4 von 544 Bestellungen, also einer Rate von unter 1 Prozent. Obwohl der Benutzer an dieser Stelle einen Mehraufwand hat, ist dieser im Vergleich zur manuellen Bestellabwicklung sehr gering, da bei dieser alle Bestellungen vom Benutzer bearbeitet werden müssen.

Um die dadurch entstehende manuelle Arbeit für den Benutzer weiter zu reduzieren, dient das Lernsystem. So müssen für nicht erkannte Bestellungen die Werte nur einmalig

erfasst werden und können in Zukunft von der Software selbstständig zugeordnet werden.

## 5.2 Lernsystem

So wurden in einem weiteren Versuchsaufbau zufällige Bestellungen in das Versandsystem importiert. Bei den Bestellungen handelte es sich um reale Bestellungen die über den Shop im Fallbeispiel ausgelöst wurden. Hierbei ist zu erwähnen, dass der Shop zu dem Zeitpunkt ein Spektrum von über 2.500 Produkten hatte, aus denen sich die Bestellungen zusammensetzten.

Tabelle 5.2 zeigt die Anzahl der Erkennungen über das Lernsystem. Der Tabelle ist zu entnehmen, dass die Erkennungsrate mit zunehmendem Datenbestand immer weiter ansteigt.

Dem stetigen Anstieg der Erkennungsrate wirken Änderungen der Produktdaten entgegen. Werden Abmessungen eines Produktes im Warenwirtschaftssystem geändert, so werden alle Bestellungen, die dieses Produkt enthalten, als Vorlage für das Lernsystem ausgeschlossen. Gleiches gilt auch für Änderungen an den Maximalwerten der Versanddienste. Auch Produkterweiterungen, durch die sich neue Bestellkombinationen ergeben können, führen zu einer geringeren Erkennungsrate. Bei den in Tabelle 5.2 vorliegenden Werten hatten diese Faktoren jedoch keinen Einfluss.

Bestellungen	davon erkannt	Erkennungsrate
55	1	1,82%
112	8	7,14%
162	15	9,26%
205	21	10,24%
260	29	11,15%
316	42	13,29%
362	56	15,46%
370	58	15,68%
433	80	18,48%
479	96	20,04%
501	103	20,56%
544	130	23,90%

Tabelle 5.2: Erkennungen über das Lernsystem

## 5.3 Speicherauslastung

Eine weitere Schwachstelle des Brute-Force-Algorithmus ist der hohe Speicherverbrauch [ZZZ07]. Die in Tabelle 5.3 dargestellten Werte entsprechen der maximalen Speicherauslastung, die bei der Berechnung einer Bestellung mit der jeweiligen Anzahl an Produkten aufgetreten ist. Die Anzahl der Messpunkte ist bei diesem Versuchsaufbau sehr gering, da die Speicherauslastung bei unter 7 Produkten nicht zu messen war und die Auslastung bei 12 Produkten bereits 1 Gigabyte überschritt.

Produkte	Speicherauslastung
7	1,84 Megabyte
8	8,59 Megabyte
9	25,06 Megabyte
10	64,64 Megabyte
11	158,38 Megabyte
12	> 1024 Megabyte

Tabelle 5.3: Speicherauslastung

Dennoch erkennt man den drastischen Anstieg der Auslastung. Dieser entsteht durch die immer größere Anzahl an möglichen Kombinationen bei der Erkennung.

## 5.4 Arbeitersparnis

Um auch den effektiven Nutzen des Versandsystems zu zeigen, wurden 50 Bestellungen verarbeitet. Zur Messung des Arbeitsaufwands dienen hierbei die minimal notwendigen Klicks, um den Versand zu bearbeiten. In Tabelle 5.4 wurden die Benutzereingaben gegenübergestellt. Hierbei wurde davon ausgegangen, dass alle erforderlichen Programme bereits geöffnet sind. Ein Datenimport ist nur im Versandsystem nötig. Hierbei fallen jedoch nur 2 Klicks an.

Bei der Zuordnung der Daten übernimmt das Programm den größten Teil automatisch. Im Versuch musste 1 Produkt nachträglich bearbeitet werden, daher mussten hier 5 Eingriffe vom Benutzer erfolgen. Bei der manuellen Zuordnung musste das Menü aufgerufen werden sowie alle 50 Bestellungen per Hand zugeordnet werden.

Der Export zu den Lieferdiensten muss in *DreamRobot* für jedes Versandunternehmen manuell gestartet werden. Im Versandsystem genügt an dieser Stelle ein Klick.

Da alle Versandmodule zum Testzeitpunkt noch auf der Erstellung von CSV-Dateien basierten, ist die Ersparnis bei den Weiterverarbeitenden Programmen sehr gering. Hier fallen nur Angaben weg, die das Versandsystem automatisch berechnen kann. Dazu zählt zum Beispiel das Gewicht der Pakete.

Für den gesamten Vorgang ergibt sich damit eine Arbeitersparnis von etwa 65 Prozent. Durch die Verwendung von Versandmodulen lässt sich hier gerade im Bereich der externen Programme noch Arbeitsaufwand einsparen. Im Anwendungsfall wurde bereits eine Schnittstelle zu DHL implementiert, durch die die Benutzereingaben in dem Bereich komplett wegfallen. Zur praktischen Anwendung dieser Schnittstelle ist jedoch eine kostenpflichtige Abnahme durch DHL erforderlich. Daher konnte diese Schnittstelle bei den Tests noch nicht verwendet werden.

	manueller Versand	Versand- system
Datenimport	0	2
Zuordnung	55	5
Export zu den Lieferdiensten	9	1
externe Programme	80	50
<b>Summe</b>	<b>144</b>	<b>58</b>

Tabelle 5.4: Klicks zur vollständigen Verarbeitung von 50 Bestellungen

## Kapitel 6

# Vergleich mit anderen Versandlösungen

In diesem Kapitel werden weitere Versandlösungen vorgestellt. Ein direkter Vergleich zwischen den Systemen ist hierbei nur bedingt möglich, da alle nachfolgend vorgestellten Programme den Schwerpunkt ihres Funktionsumfangs anders legen. In der Tabelle 6.1 werden dennoch einige technische Daten dieser Versandlösungen gegenübergestellt.

	automatisches Versandsystem	DHL Intraship	HVS32
Datenimport	automatisch	manuell	automatisch
Etikettenauswahl	automatisch	manuell	manuell
Erweiterung möglich	ja	nein	ja (durch Hersteller)
Anwendungsart	Webanwendung	Webanwendung	Windows Anwendung
besondere Hardware- schnittstellen	nein	nein	ja (Volumenscanner, Waage)
unterstützte Versandunternehmen	3 (erweiterbar)	1	über 140
Anschaffungskosten	unbekannt	kostenlos	€ 650.00 bis € 4500.00
Packstücke pro Tag	bis etwa 20.000	unbekannt	bis zu 25.000

Tabelle 6.1: Gegenüberstellung der Systeme

### 6.1 DHL Intraship

Intraship ist eine webbasierte Anwendung zum Drucken von DHL-Etiketten. Da jedes größere Versandunternehmen seinen Kunden ähnliche Software anbietet, wie zum Beispiel *Gepard* von GLS oder *Internetmarke* von der Deutschen Post, steht *Intraship* hier stellvertretend für die hauseigenen Implementierungen aller Versanddienste. Über diese Anwendungen lassen sich ausschließlich die eigenen Etiketten drucken. Es besteht keinerlei Abgleich zwischen der Software und den Lager- und Bestelldaten des Benutzers. Daher müssen die Etiketten alle manuell angelegt werden. Die einzige Alternative bieten

CSV-Dateien, die beispielsweise von *DreamRobot* erstellt werden können. Diese können von *Intraship* eingelesen werden, beinhalten jedoch nicht alle notwendigen Daten. So wird beispielsweise nicht erkannt, ob es sich um eine Sendung per Nachnahme handelt. Die fehlenden Daten müssen dann wiederum von einem Benutzer eingetragen werden. Da die Zeit, die für die manuelle Eingabe erforderlich ist, vom Benutzer abhängt, kann hier keine Angabe darüber getroffen werden, wie hoch die Anzahl der Packstücke, die man täglich versenden kann, ist.

Während die von den Versandunternehmen angebotenen Softwarelösungen alle Optionen unterstützen, über die der Lieferdienst verfügt, musste, um eine einheitliche Implementierung der Schnittstellen automatischen Versandsystems zu gewährleisten, die Komplexität des vorgegebenen Interfaces beschränkt werden. So unterstützt die im Anwendungsfall umgesetzte Schnittstelle für DHL zum Beispiel keine Expresslieferungen. Um diese Funktionen anzubieten, muss eine separate Schnittstelle implementiert werden. Gleiches gilt für den Versand ins Ausland. Da diese Funktionen im Anwendungsfall nicht gewünscht werden, wurde auf eine Implementierung vorerst verzichtet. Die Entwicklung einer solchen Anbindung ist jedoch mit einem geringen Aufwand möglich. Der Aufbau dieser Schnittstellen ist der bereits entwickelten DHL-Schnittstelle sehr ähnlich.

Obwohl es mit *Intraship* möglich ist gültige Etiketten für den Paketversand zu drucken, bietet das Programm dem Benutzer nur wenig Vorteile. So hat der Benutzer alle Optionsmöglichkeiten aus dem Angebot von DHL. Jedoch können die Versandkosten für das Unternehmen mit diesem Programm nicht optimiert werden, da kein Vergleich mit der Konkurrenz stattfindet.

## 6.2 HVS32

Bei dem Programm HVS32 von der Heidler Strichcode GmbH handelt es sich um eine sehr komplexe Versandlösung. Es werden über 140 Versandunternehmen unterstützt. Auch die Aufteilung einer Bestellung auf mehrere verschiedene Lieferdienste ist hier möglich. Diese Aufteilung erfolgt jedoch im Gegensatz zum im Fallbeispiel implementierten System nicht automatisch. HVS32 verfügt weiterhin über Anbindungen an Warenwirtschafts- und Lagerverwaltungssysteme. Der Datenimport kann so unter anderem über einen anpassbaren CSV-Import erfolgen. Auch ein automatisches Polling, bei dem die CSV-Dateien von einem bestimmten Speicherort automatisch eingelesen werden, ist möglich. Weiterhin können Datenbanken über ODBC direkt angesprochen werden, um Daten in die Versandsoftware zu importieren.

Zusätzlich bietet HVS32 auch zahlreiche Hardwareschnittstellen. Das Gewicht eines Pakets kann direkt über eine angeschlossene Waage ermittelt werden. Es existieren weiterhin Schnittstellen zu Volumen- und Barcodescannern und speziellen Etiketendruckern. Da es sich im Anwendungsfall um eine Webanwendung handelt, ist die Anbindung von besonderer Hardware problematisch. Auch wenn es in PHP Funktionen wie die `exec()`-Funktion gibt, mit deren Hilfe es möglich ist, Konsolenbefehle auf dem Server auszuführen, besteht das Problem, dass der Server im Anwendungsfall ausgelagert ist. Dadurch ist es nicht möglich eine Verbindung zur Hardware vor Ort herzustellen.



Aus diesem Grund war die Entwicklung entsprechender Hardwareschnittstellen für den Anwendungsfall nicht erforderlich. Da weiterhin die entsprechende Hardware, wie digitale Waagen oder Volumenscanner, nicht zur Verfügung stand, konnten auch keine explorativen Versuche zur Entwicklung solcher Schnittstellen durchgeführt werden.

Doch auch diese Versandlösung bietet keine automatische Erkennung der Paketaufteilung einer Bestellung an. Bestellungen können zwar eingelesen werden, müssen jedoch immer noch manuell den Versanddienstleistern zugewiesen werden, was im gegebenen Anwendungsfall schon zum größten Teil automatisch läuft [Hei]. Auf diese Weise wird weiterhin Wissen beziehungsweise Erfahrung bei dem zuständigen Mitarbeiter benötigt. Ein weiterer Nachteil ist der Zeitaufwand, der für die manuelle Zuordnung notwendig ist. Laut eigenen Angaben ist es mit der Software dennoch möglich am Tag bis zu 25.000 Pakete zu versenden [Hei]. Eine Abschätzung auf Basis der Evaluationsdaten in der Tabelle 5.1 hat ergeben, dass mit der im Anwendungsfall implementierten Software etwa 20.000 Packstücke versendet werden können.

Thematisch hätte die Automatisierung auch als Erweiterung für HVS32 implementiert werden können. Erweiterungen für das Programm werden jedoch nur vom Hersteller angeboten. Für Dritte ist der Zugriff auf den Programmcode nicht möglich und eine API wird nicht angeboten. Der Ansatz eine Anwendung zu implementieren, die die vorhandenen Im- und Export Schnittstellen des Heidler Versand Systems nutzt, wurde aus Kostengründen verworfen.

# Kapitel 7

## Zusammenfassung

Zusammenfassend lässt sich sagen, dass die im Anwendungsfall implementierte Variante einer Versandautomatisierung auf Grund der hohen Rechenanforderungen, bei großen Bestellungen, eher für Unternehmen mit einem kleineren Bestellumfang geeignet ist. Die Software bietet auch in diesem Bereich viel Optimierungspotential, was die Berechnungszeit betrifft. Betrachtet man jedoch den Aufwand, den der Benutzer hat, so ist dieser im Vergleich zu einer komplett manuellen Bearbeitung des Versands sichtlich geringer.

Derzeit werden nur wenige Softwarelösungen mit ähnlichem Funktionsumfang angeboten. Hierbei bietet keines der untersuchten Programme die Möglichkeit einer automatischen Bestimmung der günstigsten Versandart. Zudem stehen oft auch Firmeninteressen bei den Entwicklern solcher Programme im Vordergrund. So bietet DHL beispielsweise nur seine eigenen Produkte an. Bei HVS32 werden zwar Produkte vieler Versanddienstleister angeboten, jedoch erfolgt hier keine automatische Klassifikation. Der Grund hierfür ist vermutlich bei dem hohen Implementierungs- und Wartungsaufwand durch die individuellen Versandabläufe, bei den verschiedenen Händlern, zu suchen. Zudem bestimmt im Anwendungsfall der Händler selbst das verwendete Versandunternehmen. Bei zahlreichen anderen Unternehmen ist es jedoch durchaus üblich diese Auswahl dem Kunden zu überlassen. Dies macht eine Klassifizierung unnötig, wodurch die anderen vorgestellten Versandlösungen hier besser geeignet sind.

# Literaturverzeichnis

- [BB10] Boles, D.; Boles, C.: *Objektorientierte Programmierung spielend gelernt mit dem Java-Hamster-Modell*. Vieweg+Teubner Verlag, 2. Auflage, 2010.
- [BBH<sup>+</sup>99] Biethahn, J.; Biethahn, J.; Hummeltenberg, W.; Schmidt, B.; Stahly, P.; Witte, T.: *Simulation als Betriebliche Entscheidungshilfe: State of the Art und neuere Entwicklungen*. Physica-Verlag, 1. Auflage, 1999.
- [BHA11] Bertsch, M.; Huth, N.; Arenz, R.: Netzgesellschaft: Eine repräsentative untersuchung zur mediennutzung und dem informationsverhalten der gesellschaft in deutschland. [http://www.bitkom.org/files/documents/BITKOM\\_Publikation\\_Netzgesellschaft.pdf](http://www.bitkom.org/files/documents/BITKOM_Publikation_Netzgesellschaft.pdf), 2011. [zugegriffen am 02.01.2012].
- [Hei] Heidler Strichcode GmbH: Hvs32 infomappe. <http://www.heidler-strichcode.de/www/DE/FILES/InfomappeHVS32.pdf>. [zugegriffen am 02.01.2012].
- [Lip06] Lippe, W.-M.: *Soft-Computing mit Neuronalen Netzen, Fuzzy-Logic und Evolutionären Algorithmen*. Springer Verlag, 1. Auflage, 2006.
- [Nis97] Nissen, V.: *Einführung in Evolutionäre Algorithmen.: Optimierung nach dem Vorbild der Evolution*. Vieweg+Teubner Verlag, 1. Auflage, 1997.
- [Pae06] Paetz, J.: *Soft Computing in der Bioinformatik - Eine grundlegende Einführung und Übersicht*. Springer Verlag, 1. Auflage, 2006.
- [PH08] Poetzsch-Heffter, A.: *Konzepte objektorientierter Programmierung - Mit einer Einführung in Java*. Springer-Verlag, 2. Auflage, 2008.
- [Sau04] Sauer, J.: Vorlesungsscript: Neuronale netze und fuzzy control-systeme. <http://fbim.fh-regensburg.de/~saj39122/vhb/NN-Script/script/gen/title.html>, 1991-2004. [zugegriffen am 06.01.2012].
- [Sch10] Schwarz, F.: *Verstehen Sie Ihren Verstand?: Gehirnforschung für den Alltag*. Haufe-Lexware, 1. Auflage, 2010.
- [SS] Saake, G.; Sattler, K.-U.: Vorlesungsscript: Datenbanksysteme. [zugegriffen im Wintersemester 2010/11].
- [Tho05] Thoma, M. O.: *Mac OS X 10.4 Tiger*. Carl Hanser Verlag GmbH & CO. KG, 1. Auflage, 2005.

- 
- 
- [WSW<sup>+</sup>11] Wittmann, D. G.; Stahl, D. E.; Weinfurtner, S.; Weisheit, S.; Pur, S.: Shop-systeme, warenwirtschaft und versand - so verkaufen online-händler. [http://www.ecommerce-leitfaden.de/download/studien/Shop-Systeme\\_Warenwirtschaft\\_Versand.pdf](http://www.ecommerce-leitfaden.de/download/studien/Shop-Systeme_Warenwirtschaft_Versand.pdf), 2011. [zugegriffen am 02.01.2012].
- [ZZZ07] Ziegenbalg, B.; Ziegenbalg, O.; Ziegenbalg, J.: *Algorithmen: von Hammurapi bis Gödel*. Verlag Harri Deutsch, 2. Auflage, 2007.

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur mit erlaubten Hilfsmitteln angefertigt habe.

Magdeburg, den 03. Februar 2012

Vorname Name