

Entwicklung eines semantischen  
Informationssystems zur Implementierung  
verkaufsunterstützender Methoden  
in einem Onlineshop

BACHELORARBEIT

vorgelegt am

Lehrstuhl für Information Systems Research  
Prof. Dr. Andreas Meier  
Universität Fribourg

im

November 2010

von

Janick Mischler

Betreuer

Darius Zumstein

## **Abstract**

Hersteller und Lieferanten stellen Händlern Produktlisten zur Verfügung, die diese in ihre Onlineshops importieren, um ein riesiges Angebot zeitnah und automatisiert zu bewirtschaften. Die Produktinformationen sind dabei oftmals in einer umfassenden Produktbeschreibung zusammengefasst, was die semantische Datenverarbeitung durch das System stark einschränkt. Durch einen vierstufigen Prozess können diese Daten zu wertvollen atomisierten Daten verarbeitet werden, deren Semantik dem System bekannt ist. Dies ermöglicht neue und komplexere Funktionen zur Benutzerunterstützung und zur Steigerung der Wertschöpfung eines Onlineshops wie beispielsweise Produkt-Konfiguratoren oder Empfehlungssysteme.

Diese Forschungsarbeit untersucht den Prozess zur Schürfung und Integration semantischer Daten in unterschiedlichen verkaufsunterstützenden Methoden. Die entwickelten Prototypen werden dabei in ein produktives Onlineshop-System integriert und somit unmittelbar in der Praxis erprobt.

## **Stichworte zur Kategorisierung der Arbeit**

Formalisierter vierstufiger Prozess semantischer Daten, Datenprospektion, Datenschürfung, Datenintegration, Wertschöpfung durch semantische Daten, Filtersystem, Zubehörsystem, Produktkonfigurator

## Abbildungs- & Tabellenverzeichnis

Abbildung 1	Informationspyramide.....	9
Tabelle 1	Formaler Aufbau einer typischen Produktliste.....	10
Abbildung 2	Der formalisierte Prozess.....	11
Abbildung 3	Prospektion im formalisierten Prozess .....	13
Abbildung 4	Datenprospektion .....	14
Abbildung 5	Administrationsmodul Spaltenzuordnung.....	15
Abbildung 6	Schürfung im formalisierten Prozess .....	18
Abbildung 7	Dreiteiliger Prozess der Schürfung.....	19
Abbildung 8	Produktbeschreibungen .....	23
Abbildung 9	Benutzerinterface Filter Prototyp .....	24
Abbildung 10	Datenbank-Schema Attribute.....	26
Abbildung 11	Umwandlungsinterface.....	27
Abbildung 12	Benutzerinterface Filter mit regulären Ausdrücken .....	29
Abbildung 13	Erweitertes Textattribut-Interface .....	31
Abbildung 14	Attributfilter Übersichts-Interface.....	31
Abbildung 15	Attribut-Vorschau Beispielausgabe .....	32
Abbildung 16	Integration im formalisierten Prozess .....	34
Abbildung 17	Beispiel eines Produktfilters .....	35
Abbildung 18	Funktionsablauf Filtersystem .....	36
Abbildung 19	Partitionen einer Zubehörzuordnung .....	41
Abbildung 20	Zubehör-Zuordnungsinterface .....	43
Abbildung 21	Zubehör-Auflistung eines Druckers.....	44
Abbildung 22	Zubehör-Berechnungsinterface .....	45
Abbildung 23	Wertschöpfung im formalisierten Prozess .....	48
Abbildung 24	Entwicklung Deckungsbeitrag .....	50
Abbildung 25	Preiszusammensetzung Produkt.....	51
Abbildung 26	Deckungsbeiträge und Kosten .....	54
Tabelle 2	Individualsoftware und Dateneinkauf.....	57

## Abkürzungen

bzw.	beziehungsweise
CE	Consumer Electronics (Unterhaltungselektronik)
CPU	Central Processing Unit (Hauptprozessor)
CSV	Comma-Separated Values (spezielles Datenformat)
d.h.	das heisst
FTP	File Transfer Protocol (Dateiübertragungsverfahren)
GB	Gigabyte ( $\sim 10^9$ Bytes, Mengenbegriff für Speicher)
ggf.	gegebenenfalls
ID	Identifikationsnummer
i.d.R.	in der Regel
IT	Information Technologies
mySQL	Bezeichnung für relationales Datenbankverwaltungssystem
PHP	Hypertext Preprocessor (Skriptsprache)
SQL	Structured Query Language (spezielle relationale Datenbanksprache)
TB	Terabyte ( $\sim 10^{12}$ Bytes, Mengenbegriff für Speicher)
CPU	Central Processing Unit (Hauptprozessor)

## Inhaltsverzeichnis

<b>1 Einleitung</b> .....	<b>6</b>
1.1 Problemstellung .....	6
1.2 Zielsetzung.....	6
1.3 Vorgehensweise.....	7
<b>2 Semantische Daten und formalisierte Prozesse</b> .....	<b>8</b>
2.1 Begriffsdefinition .....	8
2.2 Zweck semantischer Daten .....	8
2.3 Abgrenzung semantischer Daten und Ontologien .....	9
2.4 Formalisierter Prozess.....	10
<b>3 Datenprospektion</b> .....	<b>13</b>
3.1 Quelldaten-Import (Datenlisten).....	13
3.2 Alternative Datenquellen .....	16
<b>4 Datenschürfung</b> .....	<b>18</b>
4.1 Anforderungen an ein Verfahren .....	18
4.2 Leitgedanke der Mustererkennung.....	20
4.3 Datenschürfung mit vordefinierten Filtern.....	22
4.3.1 Prototyp.....	23
4.3.2 Funktionsweise .....	25
4.3.3 Erkenntnisse .....	26
4.4 Datenschürfung mit regulären Ausdrücken.....	28
4.4.1 Prototyp.....	28
4.4.2 Funktionsweise.....	30
4.4.3 Erkenntnisse .....	33
<b>5 Integration</b> .....	<b>34</b>
5.1 Filtersystem.....	34
5.1.1 Funktionsweise.....	35
5.1.2 Die Bedeutung semantischer Daten.....	36
5.1.3 Praktischer Nutzen .....	36
5.1.4 Ausblick.....	37
5.2 Produktempfehlungen (Zubehör) .....	38
5.2.1 Problemanalyse.....	39
5.2.1.1 Zubehör aus Produkten, die bestimmten Text enthalten. 40	
5.2.1.2 Ein Produkt einer Kategorie als Zubehör zuweisen .....	41
5.2.1.3 Produkte eines Herstellers als Zubehör zuweisen .....	42
5.2.1.4 Zubehör durch Produktattribute verknüpfen .....	42
5.2.2 Funktionsweise.....	44
5.2.3 Die Bedeutung semantischer Daten.....	46
5.2.4 Praktischer Nutzen .....	46
5.2.5 Ausblick.....	47
<b>6 Wertschöpfung der Integration</b> .....	<b>48</b>
6.1 Analyse Filtersystem .....	49
6.2 Analyse Zubehörsystem.....	49

6.3 Kalkulation.....	54
6.4 Weitere Geschäftsmodelle und Strategien .....	55
6.5 Substitutionsanalyse.....	57
<b>7 Schlusswort .....</b>	<b>60</b>
7.1 Zusammenfassung.....	60
7.2 Ausblick.....	60
7.3 Kritische Würdigung .....	61
7.4 Schlusswort.....	62
<b>Literaturverzeichnis.....</b>	<b>63</b>
<b>Anhang A.....</b>	<b>lxv</b>
<b>Anhang B.....</b>	<b>lxviii</b>

# 1 Einleitung

## 1.1 Problemstellung

Diese Forschungsarbeit beschäftigt sich mit dem Prozess der Generierung semantischer Daten in einem elektronischen Informationssystem. Hierzu zählen die drei wesentlichen Stufen der Prospektion, Schürfung und der Integration mit dem gemeinsamen Ziel, die Wertschöpfung zu steigern. Es sollen Möglichkeiten zu deren Implementierung und Umsetzung erarbeitet, analysiert und in einem Prototyp eines bestehenden Onlineshops implementiert werden.

Die Beispiele innerhalb dieser Forschungsarbeit sowie die verschiedenen Prozesse und Abläufe werden oft anhand praktischer Erfahrungen aus dem Onlineshop-Business illustriert. Das Shopsystem Electronics 3000 wurde vom Autor dieser Arbeit entwickelt und implementiert.

Es stellen sich drei wesentliche Forschungsfragen, die in dieser Arbeit beantwortet werden sollen:

Wie kann der Prozess der Generierung und Integration semantischer Daten formalisiert werden?

Welches Datenschürfungsverfahren eignet sich zur Generierung semantischer Daten aus Quelldaten?

Wie können semantische Daten in konkreten Methoden des Onlineshoppings integriert und genutzt werden?

Im Sinne der betriebswirtschaftlichen Perspektive soll zudem die Frage beantwortet werden, inwiefern durch die getroffenen semantikerunterstützenden Massnahmen ein messbarer Einfluss auf die Wertschöpfung des Onlineshops Electronics3000 erzielt werden kann.

## 1.2 Zielsetzung

Diese Forschungsarbeit soll Verfahren, Methoden und Möglichkeiten aufzeigen, aus Rohdaten semantische Daten zu generieren und diese in die Wertschöpfungskette eines Onlineshops zu integrieren. Die semantischen Daten sollen dabei automatisiert generiert und in verschiedenen praktischen Anwendungen genutzt werden.

Der Fokus liegt insbesondere auf der Schürfung und Integration semantischer Daten in den Wertschöpfungszyklus eines Onlineshops. Die Implementation soll dabei aus einem wirtschaftsinformatischen Winkel betrachtet werden und die Möglichkeiten beleuchten, die

sich daraus ergeben. Auf Implementierungsdetails wird nur eingegangen, soweit dies für das bessere Verständnis notwendig erscheint. Der Kernpunkt besteht also in den Konzepten, weniger in der konkreten programmiertechnischen Umsetzung.

### **1.3 Vorgehensweise**

Das angewandte Methodenspektrum ist eine Mischung aus argumentativ-deduktiver Analyse und Prototyping. Einerseits wird auf Basis einer rein sprachlichen bzw. argumentativen Theorie eine Formalisierung des analysierten Prozesses dargelegt, andererseits wird dieser Prozess durch die Entwicklung verschiedener funktionsfähiger Prototypen evaluiert und dessen praktische Nutzbarkeit empirisch analysiert.

## 2 Semantische Daten und formalisierte Prozesse

### 2.1 Begriffsdefinition

Der Begriff „Semantik“ ist definiert als „die Lehre von der Bedeutung von Zeichen, hauptsächlich Wörtern und Sätzen [...] und mit der Beziehung dieser Ausdrücke zu Gegenständen und Begriffen, welche sie bezeichnen“ [Weinhofer 2010, S.8]. Innerhalb dieser Forschungsarbeit wird der Begriff „semantische Daten“ als übergeordnete Bezeichnung für die einzelnen Bedeutungskomponenten einer Information verwendet bzw. insbesondere für die atomisierten Eigenschaften eines Objekts.

Die Beschreibung „15 Zoll Notebook mit 4 GB RAM“ fasst verschiedene Eigenschaften eines Objekts in einer einzigen Information zusammen. In diesem Beispiel die Bildschirmdiagonale („15 Zoll“), die Art des Produkts („Notebook“) und die Grösse des Arbeitsspeichers („4 GB RAM“). Bei der Beschreibung handelt es sich um Rohdaten, die aus einzelnen Eigenschaften besteht. Diese einzelnen Eigenschaften sind atomisiert betrachtet semantische Daten, also Daten, die eine ganz spezifische Information enthalten (z.B. die Bildschirmdiagonale).

### 2.2 Zweck semantischer Daten

Semantische Daten sind sehr bedeutsam, damit ein elektronisches Informationssystem verschiedene Objekte hinsichtlich spezifischer Eigenschaften miteinander vergleichen und beispielsweise Fragen der Kompatibilität oder Äquivalenz berechnen kann. Durch die Erweiterung semantischer Daten mit Beziehungen und Regeln zu so genannten Ontologien kann ein System zudem Datenbeziehungen erkennen und verstehen.

Die Nützlichkeit semantischer Daten soll anhand zweier kleiner Beispiele illustriert werden:

Ein Onlineshop verfügt über ein sehr tiefes Angebot an Festplatten mit verschiedener Ausstattung, Anschlüssen und Speicherkapazitäten. Aufgrund der sehr grossen Anzahl Produkte wird die Produktauflistung schnell unübersichtlich.

Kunde Meier, der eine Festplatte mit genau 1 TB Speicher sucht, kann nun auf Knopfdruck eine Auflistung aller passenden Produkte angezeigt werden.

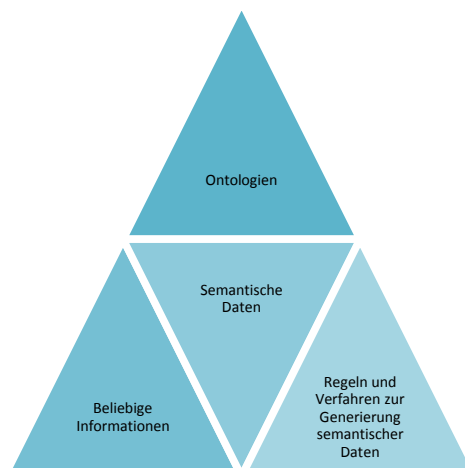
Kunde Müller, der eine Hauptplatine in den Warenkorb gelegt hat und jetzt nach einer Festplatte sucht, können automatisch Produkte angezeigt werden, deren Eigenschaften zur Hauptplatine passen, also beispielsweise nur Festplatten mit S-ATA Anschluss.

Diese Produktfilter- oder Produktempfehlungssysteme können nur zuverlässig funktionieren, wenn die notwendigen Daten in atomarer Form vorhanden sind. Auf die komplette

Produktbeschreibungen zurückzugreifen ist für das System nicht ausreichend, da es über keine menschliche Intuition verfügt und deshalb nicht erkennen kann, welche Teilinformation der Produktbeschreibung die Eigenschaft des Anschlusses oder der Speicherkapazität darstellt. Es weiss also nicht, wie Produkt A und Produkt B hinsichtlich einer spezifischen Eigenschaft (z.B. Speicherkapazität) miteinander zu vergleichen sind. Erst die Generierung semantischer Daten aus der Produktbeschreibung und deren atomisierte Speicherung nach Eigenschaft (Speicherkapazität, Anschlüsse etc.) in einer Datenbank ermöglicht beispielsweise komplexe Filter- und Empfehlungssysteme.

### 2.3 Abgrenzung semantischer Daten und Ontologien

In Bezug zum Web 3.0, dem so genannten „Semantic Web“ [HMD 271 2010], ist oft die Rede von Ontologien. Ontologien sind „sprachlich gefasste und formal geordnete Darstellungen einer Menge von Begrifflichkeiten und der zwischen ihnen bestehenden Beziehungen in einem bestimmten Gegenstandsbereich“ [Wikipedia 2010]. Im Kontext der Informatik werden Ontologien enger definiert als „[...] eine fest vorgegebene Struktur. [...] Neben individuellen Elementen enthalten Ontologien typischerweise auch Eigenschaften. Diese drücken in logischer Form ein Wissen über die Relation zwischen Elementen der Ontologien aus (dies sind typischerweise Einschränkungen über Wertekombinationen für die entsprechenden Elemente)“ [HMD 271 2010, S.7-8].



**Abbildung 1** Informationspyramide

Im Unterschied dazu stehen die semantischen Daten, die nur aus Begrifflichkeiten bzw. atomisierten Eigenschaften eines übergeordneten Objekts bestehen, aber in ihrer Entität keine Beziehungen oder Regeln enthalten. Abbildung 1 veranschaulicht den Sachverhalt anhand einer Pyramide, wobei die Komplexität der Semantik der Informatik zur Spitze hin steigt. Aus beliebigen Informationen und einer Menge passender Regeln und Verfahren

können semantische Daten generiert werden. Die semantischen Daten können wiederum mit Regeln verknüpft werden, die deren Beziehung definieren und für das System verständlich machen. Hierbei spricht man von Ontologien.

Semantische Daten sind also atomisierte Informationen, wobei Ontologien diese Informationen um Beziehungen und Zusammenhänge erweitern.

## 2.4 Formalisierter Prozess

In der Praxis stellen Lieferanten und Hersteller den Händlern Produktlisten zur Verfügung, die diese in ihre Onlineshops integrieren und so schnell ein umfassendes Sortiment aufbauen und verwalten können. Diese Produktlisten sind üblicherweise in tabellarischer Form aufgebaut, wobei jede Zeile ein Produkt und jede Spalte eine Eigenschaft des Produkts abstrahiert (die erste Zeile wird meistens für die formale Angabe der Spaltenbeschriftung verwendet und entspricht daher keinem Datensatz bzw. Produkt).

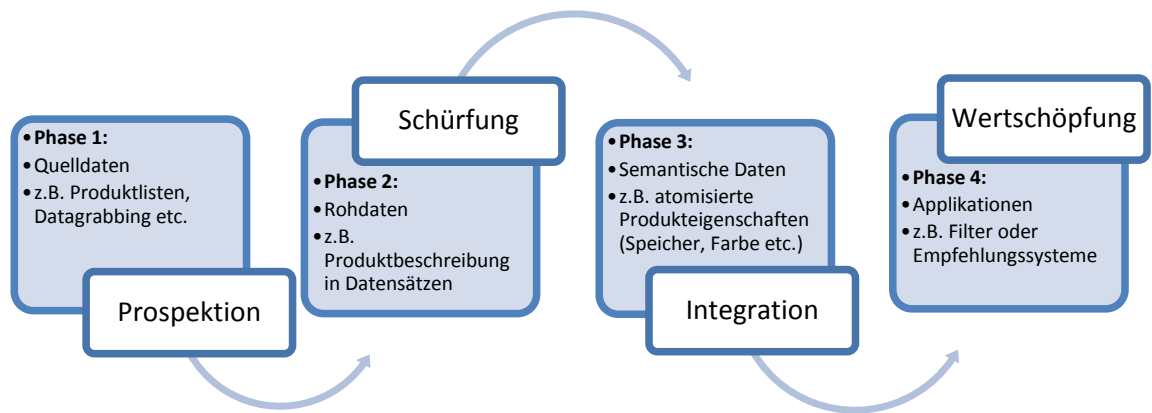
Eigenschaft 1	Eigenschaft 2	...	Eigenschaft n
Datensatz 1			
Datensatz 2			
Datensatz 3			
Datensatz 4			
...			
Datensatz n			

**Tabelle 1** Formaler Aufbau einer typischen Produktliste

Einige dieser Produkteigenschaften sind bereits atomisierte Informationen, wie beispielsweise Preis- oder Garantieangaben. Bei diesen Spalten reicht es, dem System die Spaltennummer und die Datenbankentsprechung mitzuteilen, damit es diese Daten verarbeiten, importieren und fortan darüber verfügen kann.

Die meisten Produkteigenschaften erscheinen jedoch nicht in atomisierter Form. Beispielsweise werden die technischen Produkteigenschaften oftmals in einer einzigen Spalte als Produktbeschreibung zusammengefasst. Damit kann das System in semantischer Hinsicht nichts anfangen.

Der Vorgang, aus diesen gemischten Informationen semantische Daten zu generieren, lässt sich in einem vierstufigen Prozess formalisieren. Im Unterschied zur Konzeption des „Semantic Web“ werden die bedeutsamen Daten dabei „implizit aus der vorhandenen Information“ [HMD 271, S.16] erschlossen. Es sind keine expliziten Metadaten oder Auszeichnungen der Daten notwendig. Der Benutzer definiert bestimmte Regeln zur Datenschürfung. Spezifische Integrationen der geschürften Daten führen zu Applikationen, die die Wertschöpfung steigern sollen.



**Abbildung 2** Der formalisierte Prozess

In der ersten Phase „Prospektion“ stehen die Quelldaten, die meist von einem externen Anbieter, Herstellern oder Lieferanten bereitgestellt oder bezogen werden. Hierbei handelt es sich beispielsweise um komplette Produktbeschreibungen oder Produktlisten. Diese Informationen werden in eine zentrale Datenbank importiert und dort zusammengeführt, falls beispielsweise verschiedene Produkte oder Bezugsobjekte aus unterschiedlichen Datenquellen stammen sollten.

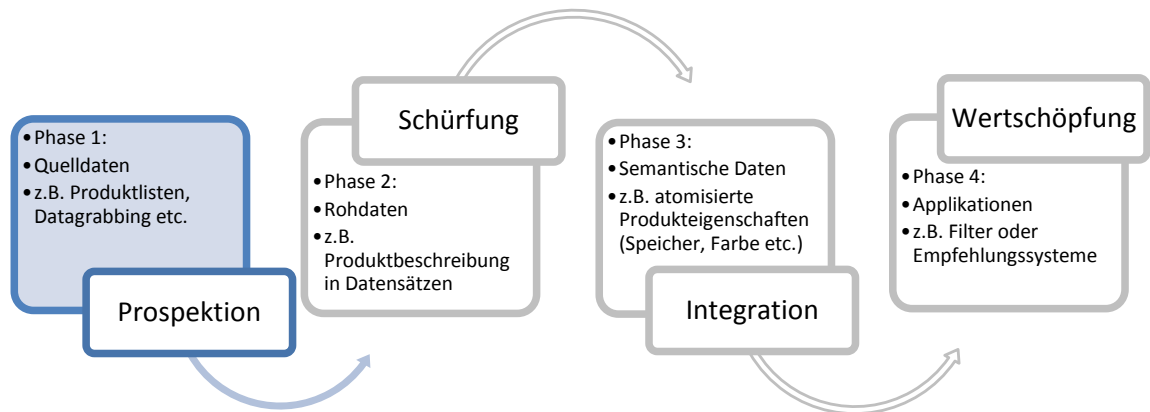
In der zweiten Phase „Schürfung“ stehen die in die Datenbank importierten Rohdaten, die nun zur weiteren Verarbeitung zentralisiert zur Verfügung stehen. Im Prozess der Schürfung werden die Rohdaten mittels spezieller Algorithmen und Verfahren verarbeitet.

In der dritten Phase „Integration“ enthält die Datenbank semantische Daten, die aus der Schürfung der Rohdaten generiert werden konnten. Die semantischen Daten sind atomisiert und nach Bedeutung referenziert gespeichert, beispielsweise die Kapazitäten der Festplatten als Zahlenwerte referenziert zu „Speicherkapazität“.

In der vierten Phase „Wertschöpfung“ werden die semantischen Daten durch verschiedene Applikationen verwendet, um beispielsweise die Benutzerbarkeit eines Onlineshops zu steigern oder dem Kunden zusätzliche passende Produkte zu empfehlen. Die Anwendung dieser Systeme entspricht im Idealfall einem Prozess der Wertschöpfung, beispielsweise in dem zusätzliche Käufe unterstützt werden oder der Kunde seine Produkte schneller und einfacher findet und den Onlineshop dadurch empfohlen wird.

Die folgenden Kapitel untersuchen die einzelnen Phasen und Prozesse dieser Formalisierung mit Schwerpunkt auf Schürfung und Integration im Hinblick auf eine Steigerung der Wertschöpfung.

### 3 Datenprospektion



**Abbildung 3** Prospektion im formalisierten Prozess

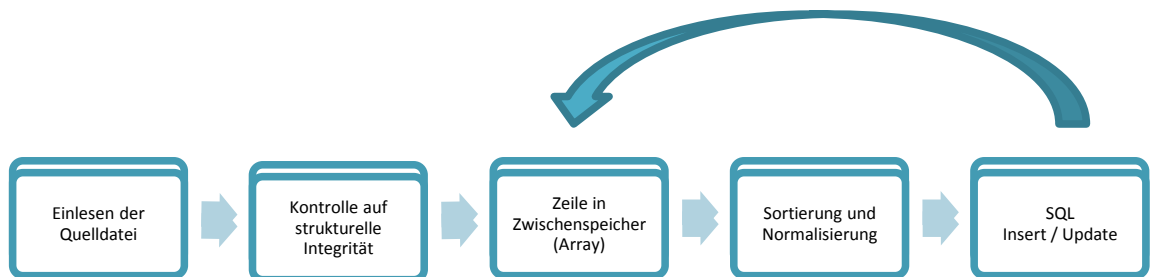
Das Kapital jedes Informationssystem sind Daten. Die Qualität von Daten kann in Anlehnung an [Gräfe 2005, S.50] bestimmt werden durch den Informationsgehalt, die Integrität, die Aktualität, die faktische Korrektheit, die Relevanz oder auch die Struktur, in der die Daten gespeichert sind. Im Kontext dieser Forschungsarbeit soll insbesondere die Struktur betrachtet werden.

Die Datenprospektion ist eine Phase des Prozesses, in dem Quelldaten zu semantischen Daten verarbeitet werden. Die Datenprospektion beschäftigt sich mit der Frage, wie Quelldaten in eine strukturierte Form gebracht werden können, um in den späteren Teilprozessen einheitlich weiterverarbeitet zu werden.

#### 3.1 Quelldaten-Import (Datenlisten)

Die Hersteller und Lieferanten stellen den Händlern oftmals einfache CSV-Dateien auf einem FTP-Server zum Download zur Verfügung. Diese bestehen aus einem Produktdatensatz pro Zeile und unterschiedlichen Produkteigenschaften pro Spalte. Die Listen sind in der Regel von Quelle zu Quelle inhaltlich unterschiedlich beschaffen, was beispielsweise den Aufbau und den Informationsgehalt eines Produktdatensatzes betrifft. Insbesondere entsprechen die verschiedenen Spalten abhängig von der Quelle unterschiedlichen Produkteigenschaften.

Damit die Quelldaten in einer relationalen Datenbank zentral und einheitlich verwaltet werden können, ist es notwendig, diese standardisiert und normalisiert zu speichern. Dadurch werden redundante Informationen und daraus resultierende Anomalien nach [Meier 2007] verhindert.



**Abbildung 4** Datenprospektion

Abbildung 4 zeigt den Prozess der Datenprospektion vom Einlesen einer Quelldatei bis zum zeilenweisen Einfügen in eine relationale Datenbank. Im ersten Schritt wird die Datei mittels einer geeigneten Sprache eingelesen, um den Zugriff sicherzustellen. Im zweiten Schritt wird die Datei auf strukturelle Integrität überprüft, d.h. die Anzahl Spalten wird mit einem für diese Quelldatei gespeicherten Schema verglichen um allfällige Veränderungen an der Struktur zu erkennen und dadurch die Integrität zu gewährleisten. Falls ein Lieferant beispielsweise eine Spalte entfernt und dies nicht rechtzeitig erkannt wird, können Daten in falsche Datenbankspalten eingefügt und dadurch falsche Angaben erzeugt werden. Die strukturelle Unversehrtheit muss zwingend erfüllt sein, damit die zeilenweise Verarbeitung der Quelldatei ablaufen kann.

Im dritten Schritt lädt das Programm jeweils eine Zeile in einen geeigneten Zwischenspeicher. Hierbei haben sich Arrays aufgrund der hervorragenden Performanceeigenschaften und dem einfachen Handling bewährt (auch wenn Arrays in typischen Webskriptsprachen wie PHP als verkettete Listen implementiert sind und damit i.d.R. ineffizienter sind als Arrays mit arithmetischem Direktzugriff wie beispielsweise in C).

Im vierten Schritt werden die Daten sortiert und für eine normalisierte Speicherung vorbereitet. Die Schwierigkeit bei Listen aus verschiedenen Quellen besteht darin, dass die Spalten mit der gleichen oder ähnlichen Eigenschaft (bzw. Produktbeschreibung oder Preis) sich an ganz anderen Stellen befinden können. Als Lösung hierzu wurde im Electronics3000 Onlineshop ein Administrationsmodul entwickelt, bei dem jeder Spalte einer Quellliste eine Datenbankentsprechung zugeordnet werden kann.

Abbildung 5 zeigt die Benutzeroberfläche des Administrationsmoduls zur Spaltenzuordnung. Die Spaltenbezeichnungen der Quelldatei werden dabei in der tatsächlichen Reihenfolge ausgegeben. Durch die Zuordnung der Datenbankentsprechung kann das Programm nun jeweils folgern, in welche Datenbankspalte die jeweilige Quellisten-Spalte eingefügt werden muss, und kann die SQL-Anweisung entsprechend generieren.

Spaltenzuordnung	Filter	Formatierung	Detaildaten	Einstellungen	FTP/HTTP
Anzahl Spaltenzuordnungen: 23					
Datenliste Spaltenbezeichnung	DB-Entsprechung	Datentitel			
Artikelnummer 2	ArtNr				
Artikelnummer	Bild1				
Bezeichnung	Produktname				
Bezeichnung 2	Kurzbeschreibung				
Lagerbestand	Lagerbestand				
Gewicht	Gewicht				
Preis (inkl. MWSt)	Einkaufspreis				
Preis (exkl. MWSt)	Preis				
WWW-Link	Deeplink				
Webtext	Prodinfortext1				
Webtext 2	Prodinfortext2				
Herstellernummer	HerstArtNr				
Hersteller	Hersteller				
MWST Satz	MwSt				
Endkundenpreis	Empf_Preis				
Garantie (Monate)	Garantie				
Erfassungsdatum	Erfassungsdatum				
Kategorie 2	Kategorie_2				
Kategorie 3	Kategorie_3				
EAN-Code	EAN				
Lieferdatum	VerfuegbarAb				
Ausverkaufsartikel	Promotion				
Discountpreis	Discountpreis				

**Abbildung 5** Administrationsmodul Spaltenzuordnung

Aus Effizienzgründen wird normalerweise zu Beginn des Vorgangs einmalig (und nicht pro Zeile) eine Vorlage für die Zieltabellen der Datenbank der SQL-Anweisung generiert und in einem Array gespeichert. So kann beim Durchlauf der Zeile jeweils das entsprechende *i*-te Element der Vorlage dem *i*-ten Datenelement der Quelldatei zugeordnet werden.

Durch dieses Administrationsmodul kann prinzipiell mit wenig Aufwand jede Datenliste in die zentrale Datenbank importiert werden, vorausgesetzt, dass diese dem Zeile gleich Datensatz und Spalte gleich Eigenschaft Schema entspricht und in der Datenbank Spalten existieren, die diesen Eigenschaften entsprechen.

Im fünften Schritt wird die generierte SQL-Anweisung ausgeführt und damit der Datensatz aus den Quelldaten in die Datenbank eingefügt. Allenfalls kann dieser Schritt derart erwei-

tert werden, dass überprüft wird, ob der Datensatz bereits existiert. In diesem Fall können auch nur einzelne Eigenschaften aktualisiert werden wie beispielsweise der Preis oder der Lagerbestand.

Die Schritte drei bis fünf wiederholen sich für sämtliche Datensätze bzw. für sämtliche Zeilen der Quelldatei. Aus Quelldaten werden so Rohdaten, die in einer einzigen zentralisierten Datenbank einheitlich strukturiert gespeichert sind.

Die Datenprospektion kann je nach Umfang der Quelldatei einige Minuten in Anspruch nehmen. Erfahrungsgemäss benötigt der Import einer Liste mit ungefähr 30'000 Produkten und integrierter INSERT-/UPDATE-Funktionalität zwischen 10 und 15 Minuten auf einem Standard-Hosting Account. Besonders zeitintensiv ist die Überprüfung auf bereits existierende Datensätze und die Ausführung der Datenbankanweisungen, die pro Zeile vorgenommen werden.

### **3.2 Alternative Datenquellen**

Daten müssen natürlich nicht zwingend nur aus gewöhnlichen Listen importiert werden. Es gibt zahlreiche weitere Möglichkeiten, automatisiert an nützliche Daten zu gelangen. So besteht eine verbreitete Variante darin, ausführliche Produktbeschreibungen aus anderen Shops zu importieren oder zu übernehmen. Diese Form der Datenbeschaffung wird als „Content Grabbing“ bezeichnet [Sury 2007].

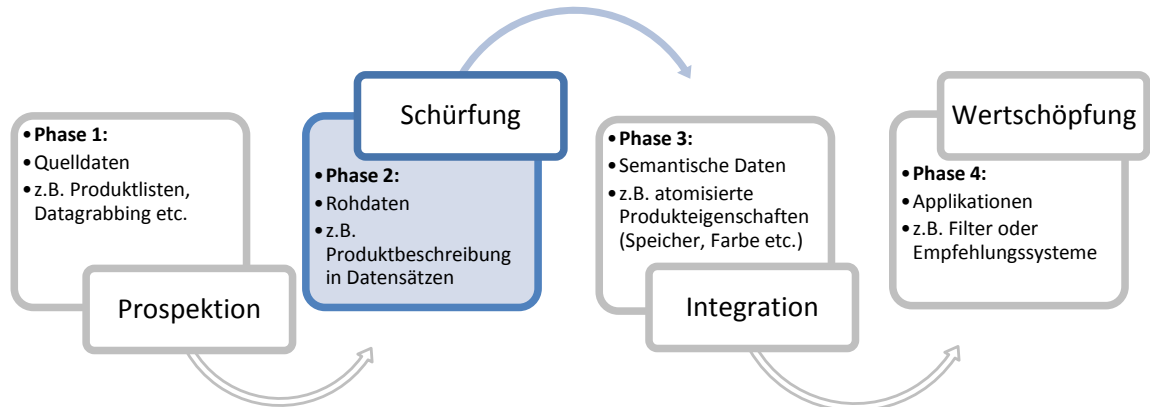
Einerseits besteht dabei aber keine Garantie hinsichtlich der Qualität der Daten, andererseits kann dieses Verfahren auch als „Datenklau“ ausgelegt werden und ist damit juristisch bedenklich. Einige Schweizer Distributoren im IT-/CE-Sektor tolerieren indes diese Form des Datenimports, zumindest bei eigenen Kunden.

Eine weitere Möglichkeit besteht darin, durch interaktive und kollaborative Elemente den Benutzer stärker in diese Phase zu integrieren, beispielsweise durch eine Kommentarfunktion oder manuell erfassbare Produktbeschreibungen. Auch diese Alternative ist allerdings als Fundament späterer semantischer Daten eher kritisch zu betrachten, da die Qualität dieser Beiträge stark schwanken kann und die Korrektheit der Daten damit nicht gewährleistet ist.

Natürlich können Daten auch auf eine Vielzahl anderer Wege und Verfahrensweisen gewonnen werden, wie beispielsweise durch die Analyse des Benutzerverhaltens auf der Seite. Diese Informationen sind zur Generierung semantischer Daten im Kontext der Produktdeskription aber im Allgemeinen nicht hilfreich (vgl. auch Kapitel 5.2.1).

Im Kapitel 6.5 werden Informationsdienstleister als Alternative zur Datenprospektion und der Datenschürfung diskutiert.

## 4 Datenschürfung



**Abbildung 6** Schürfung im formalisierten Prozess

Dieser Teilprozess beschäftigt sich mit dem Data Mining, „dem Schürfen oder Graben nach wertvoller Information in den Datenbeständen“ [Meier 2008, S.185].

Nach dem Prozess der Prospektion sind sämtliche Daten der Produktlisten in einer relationalen Datenbank gespeichert. Mittels geeigneter Algorithmen und Verfahren soll in diesen Datenbeständen nun nach semantischen Daten geschürft werden. Da das System keine Intuition hat, welche Daten wertvoll sind, müssen Methoden der Mustererkennung angewendet werden. Sobald die Daten ermittelt worden sind, werden diese gegebenenfalls überprüft und entsprechend ihrer Semantik in einer separaten Datenbanktabelle als Produktattribute gespeichert.

### 4.1 Anforderungen an ein Verfahren

Gesucht ist ein geeignetes Verfahren, das eine zuverlässige und effiziente Schürfung semantischer Daten und deren Klassifizierung nach Attributtyp erlaubt. Das Verfahren muss hinsichtlich der Komplexität gewisse Kriterien erfüllen, damit es praktikabel ist, und mittels Methoden der Datenaufbereitung sicherstellen, dass die Daten korrekt, vergleichbar und möglichst vollständig sind.

Tatsächlich ist es mittels grundlegender Algorithmen relativ unproblematisch, alle numerischen Attribute zu filtern und diese als semantische Daten zu bezeichnen. Die Kernproblematik liegt allerdings darin, dass die Daten auch klassifiziert werden müssen. Das System

muss einerseits Daten aus den Beständen schürfen, andererseits diese Daten mit einer Semantik verbinden. Alleine die Tatsache, einen dreistelligen numerischen Wert aus den Produktdaten extrahiert zu haben, lässt keinerlei Rückschlüsse auf dessen semantische Beschaffenheit zu. Es könnte sich bei diesem Wert um eine Speicherangabe handeln, aber ebenso um eine Modellbezeichnung oder eine Geschwindigkeitsangabe. Ohne eine unmittelbare Klassifizierung ist es nicht mehr möglich, die Semantik dieses Werts festzustellen. Zudem ist es wünschenswert, dass die geschürften Daten hinsichtlich der Plausibilität untersucht werden können, um quantitative Fehlerquellen möglichst zu eliminieren. Als Fehlerquelle gelten hierbei nicht nur der Schürfungsalgorithmus, sondern auch bereits die Quelldaten, die im Endeffekt irgendeinmal von fehlbaren Menschen erfasst worden sind. Es ist sicherlich besser, zu einem Produkt kein explizites Eigenschaftensattribut zu extrahieren als eines, das falsch ist.

Die Anforderungen an ein geeignetes Verfahren führen zu einem dreiteiligen, allenfalls zyklischen Prozess mit den Elementen Quelldaten, Data-Mining-Algorithmen und der Kontrolle. Als Ergebnis resultieren semantische Daten (beispielsweise Produktattribute), die grundlegend nach ihrer Bedeutung klassifiziert und normalisiert gespeichert sind, so dass semantische Daten zu den Atomen der Informationsverarbeitung werden (also nicht mehr weiter geteilt werden können, ohne einen Informationsverlust zu erleiden bzw. die Semantik zu verlieren).

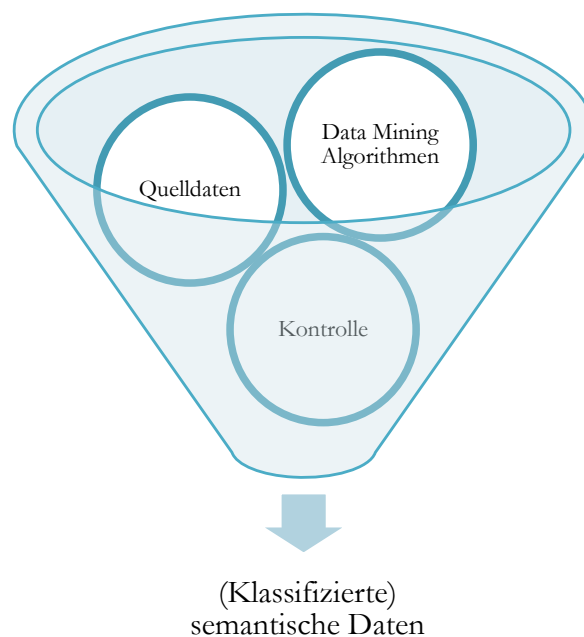


Abbildung 7

Dreiteiliger Prozess der Schürfung

Dieser dreiteilige Prozess ist bewusst als Zusammenspiel der jeweiligen Komponenten abstrahiert und nicht als lineare Abfolge. Kontrollelemente können vor, während und nach der Schürfung angewendet werden. Quelldaten wiederum können in die abschliessende Plausibilitätsprüfung einbezogen werden, um die Datenintegrität zu prüfen, und die Data-Mining-Verfahren spielen in diesen beiden Prozessen wiederum eine wichtige Rolle.

Die Qualität der resultierenden semantischen Daten muss immer auch im Verhältnis zu den investierten Ressourcen betrachtet werden. Dies ist hinsichtlich des Anforderungskatalogs wahrscheinlich die grundlegendste konzeptionelle Eigenschaft. Wird die Qualität als einziges Kriterium zur Beurteilung verschiedener Data-Mining-Verfahren betrachtet, so ist wahrscheinlich die manuelle Erfassung zu favorisieren. Eine sachkundige Person kann die technischen Daten intuitiv verarbeiten und allfällige Fehler verschiedenster Art direkt erkennen. So lange Kriterien wie Wirtschaftlichkeit, Kosten, Effizienz oder Zeit vernachlässigt werden, ist dies sicherlich die sauberste Lösung. In der Praxis muss allerdings ein Kompromiss gefunden werden, der Qualität im Verhältnis zu den anderen Kriterien fordert. So ist es wünschenswert, dass die semantischen Daten neuer Produkte möglichst zeitnah zur Verfügung stehen, ebenso sollten Beschreibungsaktualisierungen möglichst sofort in den Attributen berücksichtigt werden. Zudem sind die Kosten möglichst niedrig zu halten, um einen Wettbewerbsvorteil gegenüber der harten Konkurrenz im E-Commerce zu generieren. Daher soll innerhalb dieser Forschungsarbeit auf automatisierte, aber effiziente Verfahren fokussiert werden, die eine gute Qualität der Ergebnisse möglichst zeitnah sicherstellen.

## 4.2 Leitgedanke der Mustererkennung

Ein Vorschlag definiert Mustererkennung und Künstliche Intelligenz als „Menge aller gelösten und offenen Fragestellungen im Bereich der maschinellen Simulationen menschlicher und tierischer Intelligenzleistungen“ [Schukat/Talamazzini 2010].

Die „künstliche Intelligenz“ innerhalb dieser Forschungsarbeit fokussiert auf die maschinelle Simulation menschlicher Intelligenzleistungen, um die Mustererkennung und damit den Schürfungsvorgang zu automatisieren. Es soll also ein Verfahren entwickelt werden, das ein System erkennen lässt, welche Daten aus einer Produktbeschreibung zu extrahieren und wie diese zu klassifizieren sind.

Eine Analyse der Quelldaten zeigt, dass sich die Daten und deren Darstellung abhängig von der jeweiligen Produktkategorie stark unterscheiden. Einerseits, da die Produkte auf-

grund ihrer unterschiedlichen Natur über ganz verschiedene Eigenschaften und Attribute verfügen. Andererseits, da die Datensätze von verschiedenen Personen erfasst worden sind. Diese Feststellung bedeutet, dass es kaum möglich ist, einen effizienten, generischen Algorithmus zu entwickeln, der alle semantischen Daten aus allen Datensätzen extrahiert und klassifiziert. Ein entsprechender Algorithmus müsste auf die verschiedenen Darstellungsformen und Informationsrepräsentationen Rücksicht nehmen und zahlreiche verschiedene Musterprüfungen vornehmen, um nur das anzuwendende, korrekte Muster zu finden. Daraus resultiert ein sehr umfangreicher Algorithmus mit allenfalls ausufernder Komplexität und unsicherer Qualität.

Die Analyse zeigt aber auch, dass innerhalb der gleichen Produktkategorie die Informationen sehr oft sehr ähnlich dargestellt werden. So ist beispielsweise jede externe Festplatte mit einer Angabe über die Speicherkapazität versehen und jeder Fernseher mit einer Angabe über die Bilddiagonale. Natürlich existieren in vielen Kategorien auch eingegliederte Zubehörprodukte, die nicht über entsprechende Eigenschaften verfügen. Diese sollten mit geeigneten Methoden erkannt und entsprechend verarbeitet werden können.

Einige Beispiele von Produkttexten verschiedener Kategorien:

#### **Kategorie „Externe Festplatten“**

HD Seagate Desktop External 3.5" 500GB, 7200rpm, 16MB Cache, 11ms, USB 2.0, extern  
HD LaCie D2 Disk Quadra v2.1 3.5" 2TB, USB 2.0 FW400/800 eSATA, 7200rpm, silber  
HD Seagate Momentus 5400,3.5, 500GB, SATA2

#### **Kategorie „LCD-/Plasma-TVs“**

TOSHIBA 46SV685D, 46" LED-TV, 16:9

Premiumgarantie Home Electro

Es ist offensichtlich, dass sich die Produkte in „Externe Festplatten“ hinsichtlich der Bedeutung der Attribute ähneln. So verfügen alle drei Produkte über Angaben zur Kapazität, zur Grösse und zur Anschlussmöglichkeit. Eine gemeinsame Attributbasis ist eine unabdingbare Grundlage, um die Vergleichbarkeit (und damit nützliche Anwendungen überhaupt) sicherzustellen.

Der Auszug aus der Kategorie „LCD-/Plasma-TVs“ zeigt das Problem der implizit zugeordneten kategorienfremden Produkte. So verfügen zwar die meisten TVs über eine Angabe zur Display-Art, der Bilddiagonale und der quantitativen Bildqualität, nicht allerdings die

Produkte wie beispielsweise Garantieverlängerungen. Hier ist die Einplanung einer geeigneten Methode notwendig, um keine falschen oder leere Attribute aufgrund von Zubehör zu generieren. Andererseits ist es natürlich reizvoll, diese Zubehörprodukte in einer separaten Rubrik dem Kunden anzubieten, um im Sinne des Cross-Marketings den Umsatz zu erhöhen.

Die im Folgenden entwickelten Verfahren müssen also einerseits berücksichtigen, dass abhängig von der Kategorie unterschiedlich angepasste Mustererkennung angewendet wird aufgrund der verschiedenen Attribute. Andererseits können Kategorien einzelne Produkte enthalten, die den Charakter von Zubehör haben und dementsprechend keine typischen Attribute aufweisen.

### 4.3 Datenschürfung mit vordefinierten Filtern

In einem ersten Prototyp soll eine Datenschürfungsmethode entwickelt werden, die für verschiedene Produktkategorien einen einfach und schnell zu konfigurierenden Filter bereitstellt.

Dieser Filter basiert auf der in Kapitel 4.2 ausgeführten Annahme, dass sich alle Produkte einer Kategorie ähnliche Informationsstrukturen teilen. So verfügt jedes Notebook über eine Angabe zur Bilddiagonalen und jede Festplatte über eine Angabe zur Kapazität.

Ein Vergleich zwischen den typischen Informationsstrukturen einer Produktbeschreibung dreier wichtiger Kategorien zeigt, dass die numerischen Attribute in der Regel gefolgt sind von der klassifizierungstypischen Einheitenbezeichnung.

Als Stichprobe zeigt Abbildung 8 den Auszug des Produkttexts einer externen Festplatte, eines TVs und eines Notebooks. Die numerischen Attribute wie Speicherkapazität, Bilddiagonale oder der Arbeitsspeicher werden von typischen Einheitenbezeichnungen (GB, Zoll, GB) gefolgt. Gleichzeitig handelt es sich bei diesen Angaben um die wesentlichen Attribute zur Klassifizierung der Produkte der jeweiligen Kategorie.

Ein Kunde, der nach einem geeigneten Fernsehgerät sucht, hat wahrscheinlich konkretere Vorstellung von der gewünschten Bilddiagonalen als dem Kontrast. Dies gilt insbesondere für Normalbenutzer, also technisch weniger versierte Personen, die wiederum einen quantitativ überragenden Teil der Zielgruppe (und effektiven Kundengruppe) ausmachen.



**Abbildung 8** Produktbeschreibungen

Es bietet sich also an, ein relativ einfaches Filtersystem zu entwickeln, das die wichtigsten Attribute extrahiert, in dem es sich zu Nutze macht, dass diese in der Regel vor oder nach einer Einheitenbezeichnung stehen (und dies pro Produktkategorie meistens einheitlich ist).

### 4.3.1 Prototyp

Abbildung 9 zeigt das Benutzerinterface der entwickelten Prototyps.

In einem ersten Schritt wird die Art des Filters („Vordefinierte Filter“) sowie die Attributkategorie oder Klassifizierung der Attribute festgelegt. Pro verschiedenen Attribut muss mittels dieses Interfaces jeweils ein Filter erstellt werden, der definiert, wie dieses Attribut aus der Produktbeschreibung generiert werden kann und zu welcher Attributkategorie es gehört. Um die Festplattenkapazität und die Bilddiagonale eines Notebooks zu ermitteln, müssen also zwei Filter angelegt werden. Die Attribut-Kategorien können in einer separaten Ansicht erstellt und bearbeitet werden. Diese werden in einer eigenen Datenbank-Tabelle normalisiert gespeichert und mit den zugehörigen Attributen referenziert.

Im Register „Kategorien-Mehrfachauswahl“ erfolgt eine Auswahl der Kategorien, auf die dieser Filter angewendet werden soll. Da sich beispielsweise in mehreren Kategorien Festplatten befinden (externe Festplatten, S-ATA Festplatten, SSD-Festplatten etc.), die eine strukturell ähnliche Produktbeschreibung zeigen, ist es sinnvoll, einen Filter gleich auf mehrere passende Kategorien anwenden zu können.

Im Register „Attribut-Filtereinstellungen“ werden die Parameter für den Filter festgelegt. Dazu gehört der finale Attributstring, also der Text, der hinter jedem generierten numerischen Attribut angezeigt werden soll. Der Suchstring, nach dem in der Produktbeschreibung gesucht wird und optional die Anzahl Zeichen, innerhalb derer ausgehend vom Suchstring ein Attribut generiert werden soll. Die „Anzahl Zeichen“-Option ist allerdings nur nützlich, falls ein Attribut vor oder nach einer bestimmten Zeichenkette steht und immer gleich lang ist.

**Filtereinstellungen für Attribute**

**Filter auswählen:**  
Vordefinierte Filter

**Filterparameter erfassen:**  
Attribut-Kategorie wählen: Festplatten-Speicher

+ Kategorien-Mehrfachauswahl

+ Filter-Information

+ Attribut-Filtereinstellungen

Filter aktivieren:

Finaler Attributstring: GB

Suchstring: GB

Anzahl Zeichen:

Numerisches Attribut suchen?:

Suchstring:  vorher  nachher

+ Umwandlungen

+ Datenbank-Einstellungen

In welcher Datenbank-Tabelle suchen?: Produkt

In welcher Datenbank-Spalte suchen?: Produktname

Filter speichern (Vorschau)

**Abbildung 9** Benutzerinterface Filter Prototyp

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), Administrationsbereich, 2010)

Falls die Checkbox „Numerisches Attribut suchen?“ aktiviert wird, sucht der Filter vor oder nach (entsprechend der Auswahl) dem gefundenen Suchstring nach dem ersten numerischen Attribut. Im Register „Datenbank-Einstellungen“ wird festgelegt, auf welche Spalte der Datenbank mit den Produktinformationen der Filter angewendet werden soll.

Mit diesem Prototyp ist es möglich, Attribute bzw. semantische Daten zu generieren, unter der Bedingung, dass diese über eine Produktkategorie betrachtet durch die Kombination aus einem konstanten Suchtext und einem vorhergehenden oder nachfolgenden numerischen Attribut eindeutig beschrieben werden können.

### 4.3.2 Funktionsweise

In der relationalen Datenbank existiert eine Tabelle, die sämtliche Einstellungen des Filters speichert. Dadurch ist es möglich, einen Filter nicht nur einmal, sondern automatisiert immer wieder auf die veränderten Produktdatensätze anzuwenden. Dies ist wichtig, da an jedem Arbeitstag eine Produktliste mit neuen Datensätzen importiert wird, deren semantische Daten ebenso extrahiert werden sollen wie die der anderen Produkte in derselben Kategorie.

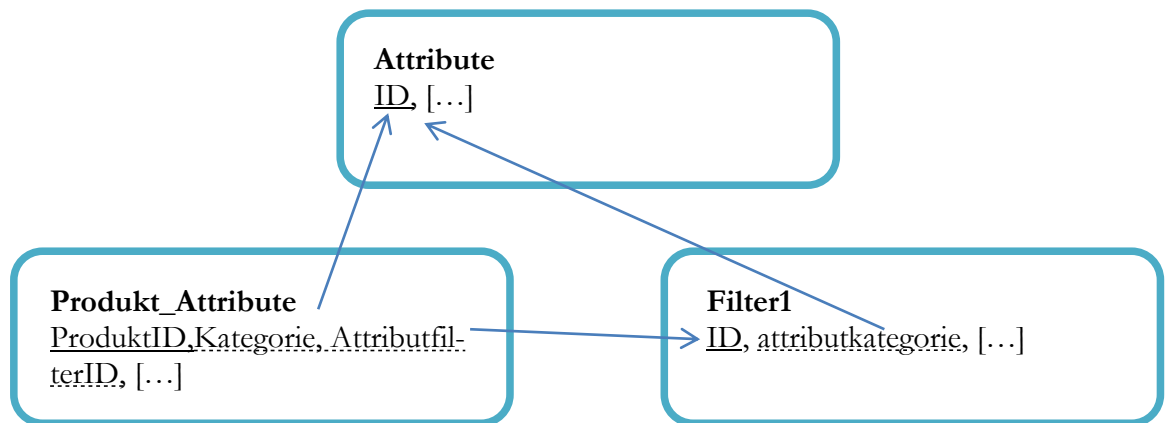
Beim Ausführen des Filters werden sämtliche generierten semantischen Daten in der Tabelle „Produkt\_Attribute“ gespeichert. Diese enthält eine Referenz auf die Identifikationsnummer des Produkts, das numerische Attribut, den Attributtext („Finalstring“), eine Referenz auf die Identifikationsnummer der Attributkategorie (Klassifizierung) sowie optional einen Statuswert und die Identifikationsnummer des attributerzeugenden Filters. Die Widerspruchsfreiheit der Datensätze wird durch geeignete SQL-Methoden der deklarativen Integrität (UNIQUE-Klausel für Produkt ID und Attributkategorie ID) sichergestellt. Dies ist notwendig, damit Datensätze einfach und effizient aktualisiert werden können durch Überschreiben und damit keine doppelten Attribute der gleichen Attributkategorie für dasselbe Produkt auftreten (vgl. Abbildung 10).

Jedem Filter muss eine bestimmte Attributkategorie zugewiesen werden, damit das System nicht einfach Daten atomisiert, sondern auch die zugehörige Semantik bzw. Kategorie referenziert. Nur so ergeben sich semantische Daten.

Die Attributkategorien werden in der Tabelle Attribute gespeichert. Diese enthält Spalten für die Identifikationsnummer der Kategorie, deren Bezeichnung, eine Sortiernummer, eine Referenz auf einen Glossareintrag (beispielsweise um Erklärungen zu Fachbegriffen anzubieten) sowie eine numerische Angabe zum Ausgabebetyp (beispielsweise um Daten zu runden oder mit x-Nachkommastellen auszugeben).

Abbildung 10 zum Datenbank-Schema illustriert die Beziehungen zwischen den drei Datenbank-Tabellen „Attribute“ (enthält Attributkategorien), „Produkt\_Attribute“ (enthält die extrahierten Attribute) und „Filter1“ (enthält die erstellten Filter).

Eine Anwendung kann jetzt auf ein beliebiges Attribut eines Produkts zugreifen, in dem sie alle Datensätze der Tabelle „Produkt\_Attribute“ mit der entsprechenden Produkt ID und der ID der Attributkategorie auswählt. Da es pro Produkt und Kategorie nur jeweils ein Attribut geben darf (UNIQUE-Klausel), enthält das Resultat einen oder keinen Datensatz. Die Bedeutung des Attributs kann aus der referenzierten „Attribute“-Kategorie gelesen werden.



**Abbildung 10** Datenbank-Schema Attribute

Die Implementation dieses Filter-Prototyps wurde in der Skriptsprache PHP mit einzelnen Java Script Komponenten realisiert. Damit der Filter aus den Daten Attribute gemäss den definierten Regeln finden kann, wird in einem Kontrollmodul entsprechend der Filterparameter eine Abfolge von Funktionen zur Verarbeitung von Zeichenketten (Strings) ausgeführt.

### 4.3.3 Erkenntnisse

Dieser Prototyp ist in der Praxis sehr nützlich, um Attribute aus einfachen Mustern einer vorgegeben strukturellen Form zu extrahieren. Wie bereits in den Vorkapiteln ausgeführt, kann dieser Filter nur numerische Attribute erkennen, die sich vor oder nach einer konstanten und eindeutigen Zeichenkette befinden. Er ist aber nicht fähig zu erkennen, ob eine numerische Angabe wirklich dem gewünschten Attribut entspricht. Ebenso scheitert dieses Verfahren an komplexeren Mustern, die auf nicht-konstanten Suchtexten basieren. Bereits die Anforderung „Die Kapazität steht jeweils vor oder nach der Zeichenkette ‚GB‘“ überfordert das System bzw. benötigt zwei Filter. Zudem ist die Datenintegrität nicht gewährleistet, da es denkbar ist, dass beispielsweise vor der Zeichenkette gesucht und ein

numerischer Wert gefunden wird, der aber nicht der Speicherkapazität entspricht, sondern einer anderen Eigenschaft.

Ebenso zeigt sich die Schwierigkeit, dass Attribute in verschiedenen Masseinheiten angegeben werden können. Eine Festplatte mit 1024 GB Speicher entspricht einem Terabyte oder TB. Hierzu sollte es eine Möglichkeit geben, verschiedene akzeptierte Masseinheiten für den gleichen Filter zu definieren und die numerischen Werte entsprechend umzurechnen. Eine entsprechende Möglichkeit wird im Register „Umwandlung“ implementiert.



The image shows a user interface for conversions. It has a title bar with a plus sign and the text 'Umwandlungen'. Below the title bar, there are three rows of controls:

- The first row is labeled 'String umwandeln:' and has a text input field containing the number '0'.
- The second row is labeled 'Umwandlungsfaktor:' and has a text input field containing the number '0'.
- The third row is labeled 'Umwandlung aktivieren?:' and has a checkbox that is currently unchecked.

**Abbildung 11** **Umwandlungsinterface**

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), Administrationsbereich, 2010)

Hierbei kann für „String umwandeln“ eine zusätzliche Zeichenkette angegeben werden, nach der gemäss den Filterparametern gesucht und deren zugehöriger numerischer Wert mit dem gewünschten Umwandlungsfaktor multipliziert wird.

Beim Festplatten-Beispiel könnte als „String umwandeln“ also „TB“ eingegeben werden und als Umwandlungsfaktor „1024“, so dass alle Terabyte-Angaben in Gigabyte umgerechnet werden und so alle Attribute der Kategorie „Speicherkapazität“ in der gleichen Masseinheit gespeichert sind.

Zusammenfassend gilt es also festzuhalten, dass die präsentierte Filtervariante eine Möglichkeit bietet, aus relativ standardisierten Produktinformationen numerische Attribute zu generieren. Diese Methode ist allerdings zu wenig flexibel und dynamisch, da sie die Mustererkennung auf dieses eine statische Modell einschränkt, dafür aber einfach zu bedienen, schnell und zuverlässig in der Ausführung (bei konstanter Datenstruktur). Allerdings lässt sie keine Qualitätskontrolle zu und reagiert relativ anfällig auf Änderungen in der Datenstruktur, beispielsweise wenn die Zahl vor statt hinter der angegebenen Zeichenkette steht.

In der Praxis ist diese Variante ausreichend, um in den populärsten und standardisierten Kategorien einige Attribute zu generieren und den Kunden eine einfache Filterfunktion zur Verfügung zu stellen. Sie reicht jedoch nicht aus, um beispielsweise zuverlässige Produkt- oder Zubehöreempfehlungen durchzuführen oder einen Produktkonfigurator zu ermöglichen.

#### 4.4 Datenschürfung mit regulären Ausdrücken

In einem zweiten Prototyp soll eine flexible und mächtige Methode zur Schürfung wertvoller Daten aus den Quelldatenbeständen realisiert werden. Hierbei wird verstärkt auf eine generische und damit flexible Lösung fokussiert, die in einer einzigen Implementation die Erstellung von Filtern zur Schürfung sämtlicher musterbasierter Attribute ermöglicht.

Hierzu wird ein System entwickelt, das auf den konzeptionellen Angaben der vorhergehenden Kapitel beruht (d.h. Filter pro Kategorie etc.), aber auf der Methodik regulärer Ausdrücke aufbaut.

Ein regulärer Ausdruck bezeichnet „eine Methode zur Beschreibung eines Musters in einem Textstück“ (S.153, PHP5&MySQL Kompendium). Hierzu sind eine Anzahl an syntaktischen Elementen definiert, die sich zu Ausdrücken kombinieren lassen, um ein Muster oder eine Regel zu spezifizieren, auf das eine Zeichenkette geprüft wird. Zudem können Intervalle definiert werden, die den Plausibilitätsbereich der gesuchten Attribute definieren. Beispielsweise liegt der gegenwärtige Sollwert der Festplattenkapazität in Gigabyte zwischen einer zwei- und vierstelligen Ganzzahl, währenddessen der Sollwert in Terabyte ein bis höchstens zweistellig sein dürfte.

##### 4.4.1 Prototyp

Das Ziel dieses Prototyps ist die effiziente und zuverlässige Schürfung semantischer Daten mit einhergehender Plausibilitätskontrolle. Die Methode soll flexibel sein und sich auf sämtliche Informationsstrukturen innerhalb der Produktbeschreibungen derart anpassen lassen, dass sich alle einem übergeordneten Muster entsprechende Attribute zuverlässig extrahieren lassen. Diese Anforderung soll mittels regulärer Ausdrücke realisiert werden, so dass alle Attribute extrahiert werden können, ohne dass neue Funktionen programmiert werden müssen.

In einem auf diesen Filter angepassten Benutzerinterface kann nun anstelle vorgegebener Parameter eine Bedingung zur Attributgenerierung als regulärer Ausdruck definiert werden. Da sich das gesuchte Muster dadurch sehr präzise spezifizieren lässt, wird die Suche standardmässig auf die gesamte Produktinformation (und nicht eine einzelne Datenbank-Spalte wie beim ersten Prototyp) ausgedehnt. Der Zeitbedarf entwickelt sich dabei linear, da die Produktinformation jedes Datensatzes unabhängig von den anderen Datensätzen verarbeitet wird. Folglich bedeutet eine doppelte Anzahl zu verarbeitender Datensätze ungefähr die doppelte Laufzeit.

Der Prototyp wird im gleichen Interface wie beim Prototyp aus Kapitel 4.3 angezeigt, d.h. es steht eine Mehrfachauswahl an Kategorien zur Verfügung, auf die der Filter angewendet werden soll, sowie eine eindeutige Auswahl der entsprechenden Attributkategorie. Neu hingegen ist das Register „Attributfilter mit regulären Ausdrücken“ (vgl. Abbildung 12), welches eine flexible Definition des Suchmusters auf Benutzerseite (und damit ohne Eingriffe in den Quellcode) ermöglicht.

**Abbildung 12 Benutzerinterface Filter mit regulären Ausdrücken**

Auf der rechten Seite in Abbildung 12 unterstützt eine Legende der wichtigsten regulären Ausdrücke und Klassen den Benutzer bei der Musterdefinition. Der reguläre Ausdruck wird dann in das Textfeld „Attributfilter Bedingung: (zur Daten-Extrahierung)“ eingetragen.

Danach ist der gewünschte, zu extrahierende Attributtyp zu wählen. Dieser Typ kann numerisch oder Text sein. Die Unterscheidung ist wichtig, da in der Datenbank-Tabelle eine Spalte zur Speicherung von Integer sowie eine Spalte zur Speicherung von Zeichenketten existiert. Bei Integer Werten kann zudem ein Text angegeben werden, der diesen Werten in der Ausgabe allenfalls nachgestellt werden soll (beispielsweise die Einheit).

Dieser Prototyp bietet zudem die Möglichkeit, leere Attribute zu verhindern. Dies ist nützlich, falls sich implizite Zubehörprodukte in eine Kategorie eingeschlichen haben.

Im Eingabefeld „Ausdruck umrechnen:“ können die Nummern der umzurechnenden Ergebnisse der jeweiligen Teilausdrücke durch Kommata getrennt angegeben werden. Dadurch ist es möglich, beliebige Umrechnungen für beliebige Einheiten zu spezifizieren.

Entsprechend dieser Auflistung sind im Eingabefeld „Umwandlungsfaktor:“ die zugehörigen Faktoren anzugeben. Um Missverständnisse zu vermeiden, wird die Umwandlung erst durch explizites Auswählen der Checkbox „Umwandlung aktivieren?“ aktiviert.

Es ist offensichtlich, dass dieser Filter wesentlich flexibler und mächtiger ist als der im vorhergehenden Kapitel vorgestellte statische Filter, allerdings setzt die Nutzung ein gewisses Know-How in Bezug zu regulären Ausdrücken voraus.

#### 4.4.2 Funktionsweise

Eine Problematik ergibt sich daraus, dass reguläre Ausdrücke meistens zur Beantwortung der Existenzfrage eines Musters in einer Zeichenkette eingesetzt werden oder um entsprechende Teilstücke zu ersetzen. Die PHP-integrierte Funktion

```
preg_match($regexpr,$input,$output);
```

liefert allerdings die Möglichkeit, die Zeichenkette aller auf ein Muster passender Teilausdrücke zurückzuliefern. Die Variable `$regexpr` enthält den zu suchenden regulären Ausdruck, `$input` die komplette Produktbeschreibung (alle Spalten konkatinert) und `$output` die zurückgelieferten Zeichenketten der Teilausdrücke in der gegebenen Reihenfolge von links nach rechts.

Aufgrund der Bedeutung für die Funktionsweise sei diese Funktionalität anhand eines kurzen Beispiels illustriert:

Regulärer Ausdruck: `([[:digit:]]+[[:space:]])?GB`


Input: Festplatte mit 320GB

Output[0] = Input, Output[1] = 320

Im ersten Element des Arrays „Output“ wird die gesamte auf das Muster passende Zeichenkette gespeichert. In den darauf folgenden Feldern jeweils die für den jeweiligen Teilausdruck gefundene Zeichenkette. Durch diese möglichen, mehrfachen Rückgabewerte können semantische Daten bzw. Attribute generiert werden, die sich aus mehreren Teilausdrücken zusammensetzen. Beispielsweise kann es wünschenswert sein, die Zahl vor und nach dem Komma zu einem Integer zu kombinieren bzw. anschliessend durch ein eigenes Trennzeichen zu trennen. Oder es kann ein regulärer Ausdruck formuliert werden, der eine ODER-Klausel enthält, und im Endergebnis wird als Attribut die Verknüpfung aller Rückgabewerte der Teilausdrücke verwendet, und allenfalls mit zusätzlichem, beliebigen Text ergänzt.

Im Prototyp ist die Möglichkeit zum Konkatenieren (vgl. Abbildung 13) derzeit nur für Text implementiert.

Attribut-Typ:

Konkatenieren  :  Aktivieren

**Abbildung 13** **Erweitertes Textattribut-Interface**

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch) Administrationsbereich, 2010)

Der Aufbau der Datenbankstruktur entspricht grundsätzlich dem Prototyp aus Kapitel 4.3. Da die Handhabung regulärer Ausdrücke aber wesentlich schwieriger und anfälliger auf Fehler ist als die Anwendung eines vordefinierten Filtersystems, wurde bei der Entwicklung auf eine sorgfältige Modularisierung geachtet. Dadurch ist es möglich, einen Filter mit regulärem Ausdruck anzulegen und diesen danach ausgiebig in einer Vorschaufunktion zu testen. Erst durch die explizite Aktivierung wird der Filter auf die produktiven Daten und die Datenbank angewendet.

Zudem wird zu jedem erstellten Produktattribut die Identifikationsnummer des generierenden Filters gespeichert. Dadurch ist es möglich, ungewünschte oder fehlerhafte Daten mit einem einzigen Klick auf „Attribute löschen“ wieder aus der Datenbank zu entfernen (vgl. Abbildung 14).

**Attribut-Filter Übersicht**

Attribut-Kategorien-Auswahl:

Shop-Kategorien-Auswahl:

**Attributfilter 105**  
LCD/Plasma-TV => Bildfrequenz  
[Aktivieren](#) [Löschen](#) [Attribut-Vorschau](#) [Filter bearbeiten](#) [Attribute generieren](#) [Attribute löschen](#)

**Attributfilter 104**  
TFT-Flachbildschirme => Displaytyp  
[Aktivieren](#) [Löschen](#) [Attribut-Vorschau](#) [Filter bearbeiten](#) [Attribute generieren](#) [Attribute löschen](#)

**Attributfilter 103**  
TFT-Flachbildschirme => Maximale Auflösung  
[Aktivieren](#) [Löschen](#) [Attribut-Vorschau](#) [Filter bearbeiten](#) [Attribute generieren](#) [Attribute löschen](#)

**Attributfilter 102**  
AMD => Sockel  
[Aktivieren](#) [Löschen](#) [Attribut-Vorschau](#) [Filter bearbeiten](#) [Attribute generieren](#) [Attribute löschen](#)

**Abbildung 14** **Attributfilter Übersichts-Interface**

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), Administrationsbereich, 2010)

Abbildung 14 zeigt jeweils die ID des Attributfilters sowie die Produktkategorie und die Attributkategorie, die durch diesen Filter abgedeckt wird. „Attribut-Vorschau“ erzeugt eine Ausgabe der Attribute, wodurch die Funktionsweise und Korrektheit der regulären Ausdrücke einfach überprüft werden kann.

Mit der Checkbox „Aktivieren“ in Abbildung 14 kann ein Attributfilter als produktiv gekennzeichnet werden, so dass er in der Folge automatisch ausgeführt und auf die zugeordnete Produktkategorie angewendet wird.

```

Leeres Attribut:Kabelkanal Vogel's EFA 6830, 23cm
Leeres Attribut:Samsung USB Wirelesstick für Internet@TV
Quelldaten: TOSHIBA 46SV685D, 46" LED-TV, 16:9 => 200 Hz
Quelldaten: TOSHIBA 55SV685D, 55" LED-TV, 16:9 => 200 Hz
Leeres Attribut:TOSHIBA 37RV685D, 37" LCD-TV, 16:9
Quelldaten: SONY KDL-46EX500, 46" LCD-TV, 16:9, black => 100 Hz
Quelldaten: SONY KDL-37EX500, 37" LCD-TV, 16:9, black => 100 Hz
Quelldaten: LG 32LE5510, 32" Edge LED, 16:9, 24p, 100Hz => 100 Hz
Quelldaten: Samsung UE32C6710, 32" LED-TV, 16:9, 100Hz => 100 Hz
Quelldaten: Samsung UE40C6710, 40" LED-TV, 16:9, 100Hz => 100 Hz
Quelldaten: Samsung UE40C6700, 40" LED-TV, 16:9, 100Hz => 100 Hz
Leeres Attribut:Premiumgarantie Home Electro
Leeres Attribut:Standardgarantie Home Electro + Portable
Quelldaten: Samsung LE32C650, 32" LCD-TV, 16:9 => 100 Hz
Quelldaten: Samsung LE40C650, 40" LCD-TV, 16:9 => 100 Hz

```

### Abbildung 15 Attribut-Vorschau Beispielausgabe

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), Administrationsbereich, 2010)

In der Vorschau wird jeweils der Produktname ausgegeben und rechts vom Pfeil das generierte Attribut mit dem nachfolgenden Attributtext (siehe Abbildung 15). In diesem Beispiel schön zu sehen ist die Erkennung von Zubehörprodukten und damit einhergehenden Vermeidung von leeren Attributen. Ebenfalls zeigt sich auf Zeile 5 ein Produkt, das zwar einem TV entspricht, aber dennoch kein Attribut anzeigt. Dies kann passieren, falls die Beschreibung über keine entsprechende Angabe verfügt oder die Angabe nicht auf das Muster des regulären Ausdrucks passt. Desto flexibler der reguläre Ausdruck gehalten wird, desto grösser wird die Anzahl generierter Attribute im Verhältnis zur Anzahl Produkte. Einhergehend kann aber eine Steigung von Falschwerten sein, falls eine zu grosse Toleranz bzw. Abweichung zu dem zu erwartenden Muster akzeptiert wird.

Falls bei der Erstellung eines Filters mehrere Produktkategorien ausgewählt werden, wird aus praktischen Gründen für jede Kategorie ein Filter erstellt. Daher sollte, zu Testzwecken, nur eine Kategorie ausgewählt werden, damit ein allfällig falscher regulärer Ausdruck nicht in jedem Attributfilter einzeln korrigiert werden muss. Natürlich wäre es auch möglich, eine zusätzliche Ebene zu schaffen, um die Datensätze normalisiert und ohne die Redundanz des regulären Ausdrucks zu speichern. Allerdings zeigt sich, dass ähnliche Kategorien oft ähnliche reguläre Ausdrücke benötigen, aber spezifische Anpassungen erfordern können, die sich dann wiederum auf alle Kategorien auswirken würden. Zudem ist das Shopsystem auf dem Kategorienprinzip aufgebaut, d.h. jedes Produkt gehört zu einer Kategorie (kann aber in mehreren angezeigt werden) und zu jedem Produkt gehört maximal ein Attributfilter pro Attributkategorie.

### 4.4.3 Erkenntnisse

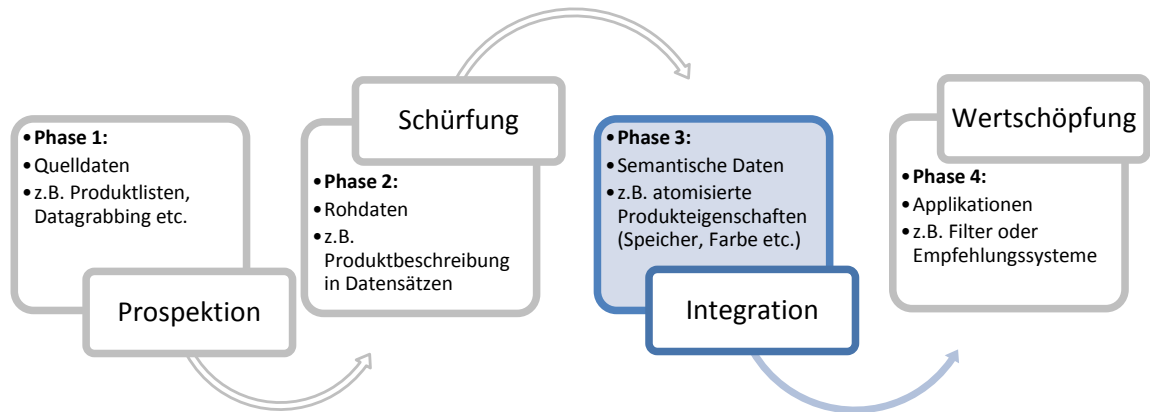
Die Methode mit regulären Ausdrücken ist weitaus flexibler und mächtiger. Damit lassen sich musterorientierte Attribute generieren, die mit der Filtermethode aus Kapitel 4.3 undenkbar gewesen wären, und das, ohne dass zur Erzeugung der Filter Eingriffe in den Programmcode vorgenommen werden müssen.

Der Nachteil liegt in der Schwierigkeit von regulären Ausdrücken, die ein gewisses Know-How voraussetzen, um effektiv, effizient und zuverlässig genutzt zu werden. Durch die Vorschaufunktion und Aktivierungsfunktion, die fehlerhafte Attribute in der produktiven Ausgabe verhindern sollten, kann der Benutzer aber wirksam unterstützt werden. Ebenso ist die Legende als nützliche Referenz praktisch.

Beim Einsatz des Prototyps hat sich auch gezeigt, dass sich einige wenige grundlegende reguläre Ausdrücke etablieren, die immer wieder benötigt werden. So muss also selten ein komplett neuer Ausdruck entwickelt werden, um die Attribute der Produkte einer Kategorie zu filtern, sondern es können oft bestehende Ausdrücke kopiert und angepasst werden.

Die Methode mit regulären Ausdrücken ist auf jeden Fall Voraussetzungen und unabdingbar, um fortschrittliche Anwendungen semantischer Daten zu implementieren. Beispielsweise ist ein Konfigurator oder ein gutes Zubehör-Empfehlungssystem ohne diese Methode kaum oder nur sehr partiell realisierbar.

## 5 Integration



**Abbildung 16** Integration im formalisierten Prozess

Dieses Kapitel präsentiert zwei verkaufsunterstützende Methoden, die durch die Verwendung von semantischen Daten zur Wertschöpfung eines Onlineshops beitragen können. Die Methoden sollen hinsichtlich ihrer grundlegenden Funktionsweise, ihres Bedarfs an semantischen Daten und ihrem praktischen Nutzen analysiert werden. Die Prototypen werden teilweise bereits seit März 2010 im Onlineshop [www.electronics3000.ch](http://www.electronics3000.ch) eingesetzt, wodurch der Nutzen und die erzielte Wertschöpfung empirisch beurteilt werden kann.

### 5.1 Filtersystem

In einem Onlineshop werden die Produkte meistens in Kategorien zusammengefasst. Jede dieser Kategorien kann prinzipiell beliebig viele Produkte enthalten, insofern die Datenbankstruktur entsprechend konzipiert worden ist. Dadurch kann die Präsentation des Angebots sehr schnell unübersichtlich und das Auffinden des gewünschten Produkts zur Tortur werden.

Ein Filtersystem kann hier Abhilfe schaffen, indem es gemeinsame Eigenschaften einer Produktkategorie erkennt und als auswählbaren Filter in der Produktaufistung anbietet.

Durch Anklicken der Checkboxen rechts in Abbildung 17 kann die Auflistung auf die Produkte mit der gewünschten Eigenschaft reduziert werden. Die Eigenschaften werden im Produktfilter nach Typ gruppiert (beispielsweise „Speicher“ und „Grösse“) und können

beliebig kombiniert werden (beispielsweise „1500 GB“ und „3.5 Zoll“). Innerhalb desselben Typs wird eine Mehrfachauswahl als logische ODER-Kombination aufgefasst.

The screenshot shows a product page for external hard drives. On the left is a navigation menu for 'PC/Laptop' components. The main content area is titled 'Externe Festplatten' and lists five products with their specifications, prices, and a 'zum Warenkorb hinzufügen' button. On the right is a 'Produktfilter' sidebar with sections for 'Preis in CHF', 'Speicher' (Storage), and 'Grösse' (Size).

Produkt	Spezifikationen	Preis
HD LaCie D2 Disk Quadra v2.1 3.5" 2TB	USB 2.0 FW400/800 eSATA, 7200rpm, silber LaCie	nur CHF 297.95 (statt CHF 299.00)
HD Freecom USB Memory 1.8" 120GB	5400rpm, 8MB Cache, 12ms, USB 2.0 Freecom	nur CHF 104.95 (statt CHF 109.00)
HD Seagate Desktop External 3.5" 500GB	7200rpm, 16MB Cache, 11ms, USB 2.0, extern Seagate	nur CHF 112.95 (statt CHF 119.95)
HD Samsung S1 250GB, 1.8	4200 RPM, 8MB, USB 2.0 Samsung	nur CHF 133.95 (statt CHF 139.00)
HD Samsung S1 250GB, 1.8", USB 2.0, blue	4200 RPM, 8MB, USB 2.0 Samsung	nur CHF 133.95 (statt CHF 139.00)

Abbildung 17

### Beispiel eines Produktfilters

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), 2010)

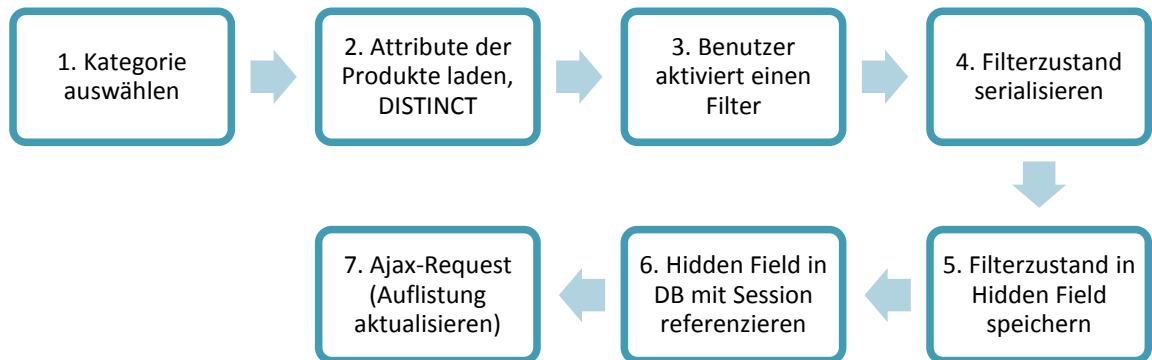
Die verschiedenen Auswahlen innerhalb der verschiedenen Typen werden durch logische UND-Kombinationen verbunden, so dass beispielsweise das Filtern nach „200 GB, 250 GB, 300 GB“ und „2,5 Zoll“ alle Produkte liefert, die „200 GB“ oder „250 GB“ oder „300 GB“ Speicherkapazität haben und von der Grösse „2.5 Zoll“ sind. Ebenfalls ist es möglich, nur Produkte in einem bestimmten Preisintervall anzuzeigen.

Ein Filtersystem bezeichnet die Eigenschaft einer Methode, eine Objektliste mit  $n$  Objekten anhand von  $m$  Kriterien aus  $k$  Kriterienklassen auf  $y$  Objekte zu reduzieren, die jeweils mindestens ein aktives Kriterium  $m_0 \in m$  jeder aktiven Kriterienklasse  $k_0 \in k$  erfüllen.

#### 5.1.1 Funktionsweise

Der Produktfilter basiert auf semantischen Daten, die der Verarbeitungsprozess generiert und in der Datenbank gespeichert hat. Sobald der Benutzer eine Kategorie in der Navigation angeklickt hat, werden alle zugehörigen Produkte aufgelistet (siehe Abbildung 17). Das Filtersystem referenziert dazu in einem ersten Schritt die ID der gewählten Kategorie mit der Produkttabelle und ermittelt alle Produkte der aktuellen Kategorie. Danach wird die ID der Produkte mit der Produkt-Attribute-Tabelle (die die atomisierten Daten speichert) und der Attribute-Tabelle (die die Bezeichnung des Eigenschaften-Typs enthält) referenziert. Der entsprechende SQL-Befehl wird mit dem Statement „DISTINCT“ ausgeführt, so dass

jedes Attribut in der Filterliste nur einmal angezeigt wird. Anschliessend erfolgt die Ausgabe der Filterattribute nach Attributtyp gruppiert.



**Abbildung 18 Funktionsablauf Filtersystem**

Sobald eine Filteroption angeklickt wird, aktualisiert das System die Produktauflistung über einen Ajax-Request. Hierzu wird die Filterauswahl in einem versteckten Formularfeld als gültiger SQL-Ausdruck gespeichert, der an das Ajax-Objekt übergeben und dort ausgeführt wird. Bei diesem SQL-Ausdruck handelt es sich um eine Serialisierung des Filtersystem-Zustands, damit dieser effizient gespeichert und wiederhergestellt werden kann. Nützlich beispielsweise, wenn ein Benutzer ein Produkt anklickt und auf die Auflistung zurückkehrt, um die Filtereinstellungen der Session wiederherzustellen.

### 5.1.2 Die Bedeutung semantischer Daten

Damit dieses Filtersystem effizient zuverlässige Produktauflistungen auf Basis der gewählten Attribute, und damit auf Basis der Bedürfnisse des Benutzers, erzeugen kann, muss es auf eine entsprechende Datenbasis zurückgreifen können. Da es sich bei den einzelnen Filteroptionen um Attribute bzw. Eigenschaften des Produkts handelt, werden dazu semantische Daten benötigt. Es ist also notwendig, dass die filterbare Produktinformation als atomare, semantische Daten zu Verfügung steht. Es ist nicht möglich, beispielsweise nur Festplatten mit einer sehr grossen Speicherkapazität anzuzeigen, wenn die entsprechenden Eigenschaften in einer ganzheitlichen Produktbeschreibung gespeichert sind, da das System die Semantik der Beschreibung nicht interpretieren kann. Die semantischen Daten sind damit in dieser Anwendung unabdingbar.

### 5.1.3 Praktischer Nutzen

Das Filtersystem ermöglicht dem Benutzer eine unkomplizierte und bedürfnisorientierte Suche in Produktauflistungen. Er gelangt dadurch schneller zum Ziel und spart wertvolle

Zeit, insbesondere wenn er sich über seine Produktwünsche bewusst ist. Durch die Ajax-Lösung entsteht der Eindruck einer in Echtzeit ablaufenden Anwendung, da das Laden der kompletten Seite bei geänderten Filtereinstellungen vermieden wird. Der Benutzer erfährt zudem eine Personalisierung des Onlineshops, da die Filtereinstellung pro Kategorie und Session in der Datenbank gespeichert werden. Schaut er sich einige Produkte an und kehrt dann auf die Produktseite zurück, werden seine Filtereinstellungen automatisch wieder geladen, was ein wesentlich komfortableres Stöbern ermöglicht.

#### **5.1.4 Ausblick**

Derzeit profitiert vor allem der zielstrebige Benutzer vom Filtersystem, der mehr oder weniger genau weiss, was er sucht. Ein gewisses technisches Know-How ist oft Voraussetzung, damit er das gewünschte Produkt findet. Ein unbedarfter Kunde kann mit den Speicherangaben möglicherweise nicht viel anfangen, da er sich nichts darunter vorstellen kann. Entsprechend wäre es eine Weiterentwicklung, einen zusätzlichen Layer über das Filtersystem zu legen und die verschiedenen Werte mit allgemeinen Qualitätseinstufungen zu abstrahieren. Beispielsweise könnten die verschiedenen Speicherangaben ersetzt werden durch drei Einstufungen „High End“, „Middle End“, „Low End“. Die Problematik besteht allerdings darin, diese Einstufungen automatisch festzulegen und an die schnelle technische Entwicklung beispielsweise gerade auf dem Hardware-Markt anzupassen. Zudem kann eine solche Abstrahierung für den technisch versierten Benutzer eine Einschränkung darstellen. Folglich müsste ein Verfahren eingeführt werden, das die Wahl der Filtervariante zulässt oder das Know-How des Benutzers automatisch einschätzt.

Falls sich sehr viele ähnliche Produktattribute ergeben, beispielsweise bedingt durch schwache Abstufungen technischer Eigenschaften, könnte die Bereitstellung einer Intervallfunktion nützlich sein. Wenn ein Kunde aus 30 Filterattributen 10 Filterattribute auswählen muss, um seine gewünschten Produkte angezeigt zu erhalten, empfindet er dies wahrscheinlich als mühsam. Sachdienlicher wäre, die ähnlichen Attributwerte in fünf oder sechs Intervallen zusammenzufassen, so dass sich das Filterangebot auf eine nützliche Auswahl beschränkt.

Eine weitere Schwierigkeit beim Filtersystem besteht darin, die richtige Anzahl an filterbaren Attributen pro Produkt zu finden und vor allem die richtigen Attribute filterbar zu machen. Im Produktfilter kann nur nach den Attributen sondiert werden, die zuvor als semantische Daten aus den Produktbeschreibungen extrahiert worden sind. Hierbei stellt sich die

Frage, welche Attribute für den Benutzer wesentlich sind und welche ignoriert werden können.

## 5.2 Produktempfehlungen (Zubehör)

In einem Onlineshop sind oft mehrere tausend Produkte gelistet. Jedes dieser Produkte gehört zu einer oder mehreren Kategorien, die wieder in Überkategorien und Sparten zusammengefasst werden. Viele Kunden erreichen das gewünschte Produkt durch „Deeplinks“ von externen Seiten oder Suchmaschinen, beispielsweise einen bestimmten Drucker. Nun wäre es betriebswirtschaftlicher Sicht wünschenswert, dass der Kunde ein USB-Kabel sowie einen Satz passender Tintenpatronen dazu bestellt. Diese Zubehörverkäufe können gesteigert werden, in dem das Shopsystem dem Kunden passende Produkte anbietet oder auf notwendiges Zubehör (beispielsweise das USB-Kabel) aufmerksam macht.

Die Frage, was ein optimales, passendes Produkt ist, versuchen zahlreiche Forschungsarbeiten zu beantworten. [Zhou 2010] hat in seiner Dissertation in theoretischer Physik an der Universität Fribourg gezeigt, dass eine Mischung aus verlässlichen und vielfältigen Empfehlungen eine langfristig optimale Variante darstellt. Einerseits werden dem Kunden demnach verlässliche, d.h. standardmässig dazu gekaufte Produkte angeboten, andererseits aber auch vielfältige, d.h. in einem etwas weiteren Rahmen gefasste, speziellere Produkte. Die verlässlichen Produkte werden dabei aus Bestellungen generiert, die Kunden mit ähnlichem Warenkorb getätigt haben. In diesem Sinn wird die „kollektive Intelligenz“ vieler Kunden benutzt, um passende Produkte aus empirischen Erkenntnissen zu berechnen.

Ein solches System ist aber gerade bei neuen Onlineshops nicht praktikabel, da die notwendigen Erfahrungswerte und Informationen fehlen. Man spricht hierbei auch vom Kaltstartproblem [Stormer 2008].

Innerhalb dieser Forschungsarbeit soll deshalb ein Prototyp eines Zubehörsystems entwickelt werden, welches auf Basis der semantischen Daten, der Produkteigenschaften und Produktattribute, sowie hinzugefügter Regeln und Verbindungen verlässliche Produktempfehlungen generiert. Demnach geschieht die Erstellung von Zubehörregeln manuell, währenddessen die entsprechenden produktspezifischen Zubehöre automatisch für eine Menge von Produkten aus diesen Regeln generiert werden.

### 5.2.1 Problemanalyse

In der Literatur werden Empfehlungssysteme oft mit dem Verfahren des „kooperativen Filterns“ gleichgesetzt und definiert [Höhfeld/Kwiatkowski 2007]. Eine solche Definition findet sich zum Beispiel in [Sorge 2010], der die Aufgaben eines Empfehlungssystems als „einem Nutzer Objekte empfehlen, die für ihn interessant sein könnten“ und „eine personalisierte Empfehlung von Objekten empfehlen“ spezifiziert. Unter dieser Definition kann ein Zubehörsystem nicht als Empfehlungssystem eingeordnet werden, da es der Anforderung der „personalisierten Empfehlung“ nicht genügt. Ein Zubehörsystem empfiehlt einem Nutzer Objekte, die für ihn interessant und vor allem kompatibel zum übergeordneten Hauptobjekt sind. Die Vereinbarkeit von Zubehör- und Hauptprodukt überwiegt dabei den Nutzen personalisierter Empfehlung. [Höhfeld/Kwiatkowski 2007] definieren Empfehlungssysteme als „Systeme, die Nutzern Empfehlungen aussprechen“ und kategorisieren diese in personalisierte und nicht-personalisierte Systeme. In Anlehnung an diese Definition lässt sich ein Zubehörsystem als nicht-personalisiertes System einordnen, das basierend auf Regeln formal für jeden Nutzer  $A$  zum Objekt  $B$  dieselbe Empfehlungsmenge  $C$  aus  $n$  Objekten erzeugt. Bei nicht-personalisierten Systemen wird das Kaltstartproblem minimiert, da die Empfehlungen regelbasiert und unabhängig von Nutzerdaten generiert werden können.

Das Hauptproblem des Zubehörsystems besteht darin, Regeln festzulegen, die definieren, welchem Produkt welches Zubehör zugeordnet werden kann. Die Regeln müssen dabei mindestens so generisch sein, dass sich aus  $m$  Regeln mindestens  $n$  Zubehörprodukte generieren lassen, wobei  $n \gg m$ . Für  $n = m$  oder sogar  $n < m$  wäre die Lösung nicht effizienter als die manuelle Erfassung und damit betriebswirtschaftlich nicht sinnvoll.

Eine Analyse der Produkte und möglichen Zubehöre zeigt, dass ein beachtlicher Teil des Zubehörs durch vier Zuordnungstypen bzw. Regeltypen definiert werden kann.

- a. Zubehör aus Produkten, die bestimmten Text enthalten
- b. Ein Produkt einer Kategorie als Zubehör zuweisen
- c. Produkte eines Herstellers als Zubehör zuweisen
- d. Zubehör durch Produktattribute verknüpfen

Alle Zuordnungstypen übernehmen das Konzept der Attributfilter, wonach ein Typ bzw. eine Regel jeweils auf eine Zielkategorie beschränkt ist. Oder anders ausgedrückt: Da sich die Produkte in den verschiedenen Kategorien meistens stark unterscheiden (und rein

funktional anderes Zubehör nachfragen), kann eine Zubehörregel nur jeweils für höchstens alle Produkte einer Produktkategorie festgelegt werden.

### 5.2.1.1 Zubehör aus Produkten, die bestimmten Text enthalten

Die Untersuchung verschiedener Kategorien und ihrer verlässlichen Zubehöre ergibt, dass sich diese häufig über zwei bis drei Schlagworte, eine Quellkategorie und ein Preisintervall zuordnen lassen. Ein einzelnes Schlagwort reicht aufgrund der Mehr- oder Vieldeutigkeit in der Regel nicht aus, um eine eindeutige Klassifizierung eines Produkts hinsichtlich seiner Semantik zu erreichen. In Versuchen innerhalb dieser Forschungsarbeit konnten mit zwei bis drei Schlagwörtern aber sämtliche erprobten Produkte zuverlässig spezifiziert werden.

Das Preisintervall als Kriterium bietet sich an, da es intuitiv korrekt scheint, dass die wenigsten Kunden ein Zubehör kaufen werden, das wesentlich teurer ist als das Hauptprodukt. Ebenso verfügt das Shopsystem über eine Kategorisierung, die einen Rückschluss auf die darin zugeordneten Produkte und deren Eigenschaften zulässt. So ist es eher unwahrscheinlich, dass sich in der Kategorie „Rasenmäher“ ein passendes Zubehör für einen Laserdrucker befindet.

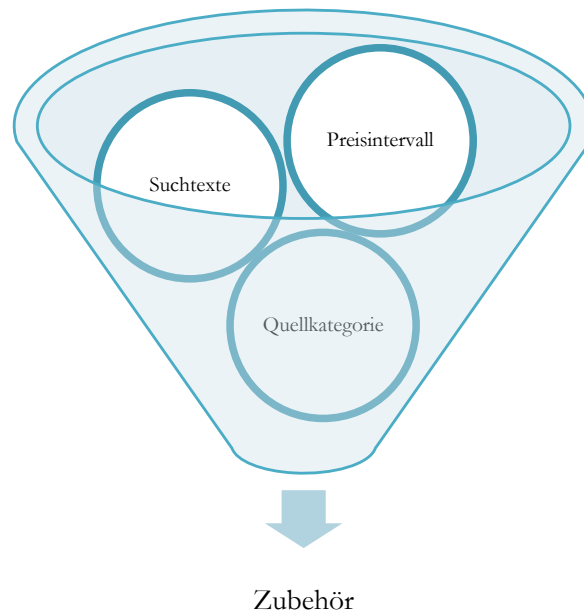
Im Beispiel der Drucker ist ein häufiges und notwendiges Zubehör ein USB-Kabel, um das Gerät an einem Computer anzuschliessen. Egal welchen Drucker man auswählt, dieses Zubehör ist prinzipiell immer passend.

Doch wie kann allen Druckern nun ein Angebot mit einigen USB-Kabeln verschiedener Länge als Zubehör zugewiesen werden? Mittels dem Zuordnungstyp a.) können neben der Quellkategorie des Zubehörs und dem Preisintervall vier Suchtexte spezifiziert werden, die in der Produktbeschreibung passenden Zubehörs enthalten sein müssen.

Im konkreten Beispiel könnten diese Suchtexte „USB“ und „Kabel“ lauten. Wissend, dass es verschiedene USB-Kabel gibt, könnten diese Begriffe um „A-B“ erweitert werden, den Typ Kabel, der mit einem Ende an Peripherie-Geräte und mit dem anderen Ende an einen USB-Port des Computers gesteckt werden kann.

Eine relativ einfache und effiziente Suche in der Produktbeschreibung aller Produkte, die sich in der angegebenen Quellkategorie und im definierten Preisintervall befinden, nach den spezifizierten Suchbegriffen, reicht hier also bereits aus, um zuverlässig passendes Zubehör zu generieren. Die Suche kann auch auf alle Kategorien ausgeweitet werden, falls keine Quellkategorie spezifiziert wird.

Diese Methode ist pragmatisch betrachtet nichts anderes als eine automatisierte Durchführung und Bereitstellung der Ergebnisse einer Suche nach gewissen Begriffen, die der Kunde auch manuell über die Suchfunktion des Shopsystems durchführen könnte. Mit dem Vorteil, dass der Kunde allenfalls zu einem spontanen Zusatzkauf veranlasst wird.



**Abbildung 19** Partitionen einer Zubehörzuordnung

Diese Regel ist im Allgemeinen effizient, da durch die Angabe von geeigneten Suchbegriffen mehrere Zubehörprodukte für mehrere Produkte generiert werden. Durch eine Regelanzahl  $m = 1$  können  $\sum_{i=1}^{AnzProd} n_i \gg m$  Zubehörprodukte generiert werden, wobei  $n_i$  die Anzahl generierter Zubehörprodukte für das  $i$ -te Produkt bezeichnet. Dabei kann es sich um Produkte desselben Produkttyps handeln, die sich aber durch eine spezifische Eigenschaft unterscheiden. Da eine geeignete Regel mindestens zwei Zubehörprodukte für mindestens ein Produkt generiert (ansonsten könnten die Zubehörprodukte direkt zugewiesen werden und die Regel wäre nicht sinnvoll), ist die Summe der Zubehörprodukte wesentlich grösser als die Anzahl der erfassten Regeln und damit gemäss der Definition (vgl. Kapitel 5.2.1) effizient.

### 5.2.1.2 Ein Produkt einer Kategorie als Zubehör zuweisen

Eine Betrachtung der Kategorie „LCD-/Plasma-TVs“ und der bisherigen Kaufentscheidungen ergibt, dass kein einziger Internet-Kunde eine Garantieverlängerung bestellt hat, dafür aber praktisch jeder Kunde, dem diese durch einen Verkäufer persönlich empfohlen worden ist. Es liegt nahe, diese Garantieverlängerungen, die für alle Fernsehgeräte und elektrischen Geräte gelten, als Zubehör anzubieten.

Dazu eignet sich die Zuordnungsregel b.), die ein einzelnes Produkt anhand seiner Identifikationsnummer (ID) jedem Produkt einer Kategorie im spezifizierten Preisintervall zuordnet.

Da sich die Garantieverlängerungen jeweils auf Produkte innerhalb eines bestimmten Preisintervalls beziehen (z.B. CHF 500 bis CHF 1000, CHF 1000 bis CHF 1500), kann die jeweilige Garantieverlängerung als Zubehör über das Preisintervall den jeweils geeigneten Produkten zugeordnet werden. Mit wenigen Klicks bietet damit jedes Produkt als Zubehör eine passende Garantieverlängerung an.

Diese Regel erlaubt also die automatisierte Zuordnung eines Produkts an eine ggf. wechselnde Anzahl Produkte einer Kategorie. Diese Regel ist effizient für eine Anzahl Produkte  $n > 1$  in der spezifizierten Kategorie und insbesondere für Kategorien mit wechselnder Produktzuordnung oder einer Anzahl Produkte  $n \gg 1$ , da dadurch die manuelle Zubehörordnung durch eine einmalig zu erfassende Regel automatisiert werden kann.

### 5.2.1.3 Produkte eines Herstellers als Zubehör zuweisen

In der ursprünglichen Kategorisierung werden verschiedene Software-Produkte in einer gemeinsamen, aufgeblähten Kategorie geführt. Die Erfahrung zeigt, dass zahlreiche Kunden gerne eine Version des Office-Pakets von Microsoft mit ihrem PC oder ihrem Notebook bestellen möchten. Um den Kunden nun die Office-Pakete als Zubehör anzubieten, lässt sich die Zuordnungsregel c.) auf die Software-Kategorie anwenden, welche alle Produkte des Herstellers Microsoft allen Produkten mit einem bestimmten Preisintervall in der spezifizierten Kategorie zuweist. Eine einzige Zuordnungsregel reicht also aus, um sämtliche Notebooks mit den Microsoft Office Anwendungen als Zubehör zu versehen.

Diese Regel ist effizient für eine Anzahl Produkte des spezifizierten Herstellers in der spezifizierten Kategorie  $n_{Hersteller} > 1$  oder eine Anzahl Produkte  $n > 1$ , denen das Zubehör zugeordnet wird. Tatsächlich zeigt sich, dass durch diese Regel oft eine variable Zubehörmenge  $A$  einer variablen Produktmenge  $B$  zugeordnet werden kann, wobei  $B \gg A$  und  $|B| \gg 1$ .

### 5.2.1.4 Zubehör durch Produktattribute verknüpfen

Die Zuordnungsregeln a.) bis c.) sind aufgrund ihrer strukturellen Einfachheit und Schlichtheit sehr effizient und zuverlässig einzusetzen. Ähnlich wie beim ersten Filterprototyp sind diesen jedoch Grenzen gesetzt, wenn es um eine bedingte Zuweisung von Zubehörprodukten geht. Beispielsweise ist es wünschenswert, dass bei jedem Drucker das passende Verbrauchsmaterial als Zubehör angeboten wird. Da die Tintenpatronen aber von

Modell zu Modell sehr unterschiedlich sind, können diese nicht automatisch über eine der bisherigen Zuordnungsregeln verknüpft werden. Weder der Preis noch der Hersteller lassen Schlussfolgerungen über die Zugehörigkeit zu. Ganz abgesehen davon, dass die Originalpatronen des Druckerherstellers meistens vergleichsweise teuer sind, und preisbewusste Kunden die „Generika“ bevorzugen.

**Neue Zubehöruordnung erfassen**

Regel wählen:

Beschreibung: Produkte per Attribut und Kategorie als Zubehör verknüpfen. Sucht in allen Kategorien der "Kategorie-Liste" in den Produktspalten nach dem Attribut der Attributkategorie "ID des Attributs" des Zielprodukts.

Ziel-Kategorie:

ID des Attributs:

Kategorie-Liste:

Min-Preis:

Max-Preis:

Zubehör-Kategorie:

**Abbildung 20** **Zubehör-Zuordnungsinterface**

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), Administrationsbereich, 2010)

Es liegt nahe, die semantischen Produktdaten zu nutzen, die im Prozess der Datenschröpfung erzeugt werden. Die Zuordnungsregel d.) generiert Zubehör durch die Verknüpfung von Produktattributen der Quellprodukte mit Produktbeschreibungen der potentiellen Zubehörprodukte.

Ein Produktattribut von Druckern ist beispielsweise das Druckermodell. Dieser kann mittels regulärer Ausdrücke im Kontext der Datenschröpfungungsverfahren generiert werden.

Die Zuordnungsregel d.) erlaubt zusätzlich die Angabe einer Liste von Kategorien, in denen nach Zubehör gesucht werden soll, sowie das übliche Preisintervall und die Identifikationsnummer der gewünschten Attributkategorie.

Die Zubehörprodukte werden nun zugeordnet, in dem für jedes Produkt der Zielkategorie das Attribut der gewählten Attributkategorie geladen und in den Produktbeschreibungen aller Produkte der spezifizierten Kategorienliste gesucht wird. Alle Produkte, die dabei gefunden werden, werden als Zubehör für das jeweilige Produkt verwendet.

Die Effizienz dieser Regel ergründet sich in der automatischen Zuordnung eines Zubehörprodukts A zu einem Produkt B, wobei die Kompatibilität unmittelbar die Beziehung Haupt- und Zubehörprodukt sichergestellt wird. Dabei ist im Allgemeinen für die Anzahl

verarbeiteter Produkte  $n$  pro Regel  $n \gg 1$  gegeben, woraus  $m$  Zubehörprodukte resultieren, wobei  $m \geq 1$ ,  $n \leq 1$ ,  $m \leq 1$  ist nicht (oder nur unter der Annahme, dass das betreffende Produkt und Zubehör häufig wechselt) sinnvoll, da in diesem Fall kein Produkt existiert oder eine Eins-zu-Eins-Zuordnung zwischen Produkt und Zubehör anzunehmen ist.

### 5.2.2 Funktionsweise

Die Implementierung eines Zubehörsystems stellt Herausforderungen hinsichtlich der Performance, Aktualität und Richtigkeit der Daten. Es muss ein Kompromiss zwischen Performance und Aktualität gefunden werden. Durch die algorithmische Korrektheit und sinnvolle (und korrekt angewandte) Zuordnungsregeln ist die Richtigkeit der Daten zu gewährleisten, d.h. nach Möglichkeit sicherzustellen, dass auch wirklich nur passendes Zubehör zu den Produkten angeboten wird (Beispiel siehe Abbildung 21).















Produktbeschreibung	Zubehör	Bewertungen	Fragen & Antworten
<b>Druckerzubehör</b> > mehr Produkte aus dieser Kategorie anzeigen			
	<b>Brother Zusatzschacht LT-5300 / 250 Blatt</b> für HL-5240 / HL-5250DN / DCP-8085DN Brother	nur CHF <b>247.95</b>	 1
	<b>Interprinting Drum Kit zu Brother DR-3100,</b> schwarz, ca. 25'000 Seiten Interprinting	nur CHF <b>119.95</b>	 1
	<b>Pelikan Toner Brother DR-3100, schwarz</b> ca. 25'000 Seiten Pelikan	nur CHF <b>127.95</b>	 1
	<b>Pelikan Toner Brother TN-3170, schwarz</b> ca. 7000 Seiten Pelikan	nur CHF <b>88.95</b>	 1
	<b>Toner Brother TN-3130, schwarz</b> HL-5240/5280, 3500 Seiten bei 5% Deckung Brother	nur CHF <b>104.95</b>	 1
	<b>Toner Brother TN-3170, HL-5240, HL5250D</b> ca. 7000s@5% Brother	nur CHF <b>129.95</b>	 1
	<b>Trommel Brother DR-3100</b> ca. 25000s@5%, HL-5240, HL5250DN Brother	nur CHF <b>196.95</b>	 1

Abbildung 21

### Zubehör-Auflistung eines Druckers

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), 2010)

Die Datenbankstruktur lässt sich mit dem Attributsystem vergleichen. Die Tabelle „Supplies“ speichert die erstellten Zuordnungsregeln. Die Tabelle „Supplies\_Titel“ enthält den Rubrikttitel, der mehreren Zubehörprodukten derselben Zuordnungsregel übergestellt wird. Die Tabelle „Produkt\_Supplies“ enthält die Produkt ID, die Supplies ID (Produkt ID des Zubehörprodukts), die ID der Zubehörkategorie sowie die ID des Rubrikttitels. Jedes Zubehör eines Produkts entspricht also einem Datensatz in der Tabelle „Produkt\_Supplies“.

Alle Zuordnungsregeln haben gemeinsam, dass sehr viele Produkte und Datensätze analysiert werden müssen. Beziehen wir uns auf Zuordnungsregel d.) und setzen wir dafür folgende Parameter voraus:

- Die gewählte Zielkategorie enthält  $n$  Produkte
- Die Attributkategorie „Druckermodell“ wird gewählt
- Das Zubehör kann in allen Kategorien gesucht werden
- Es wird kein Preisintervall definiert, da jedem Drucker unabhängig von seinem Preis das passende Zubehör zugeordnet werden soll

Für alle  $n$  Produkte wird jeweils das zugehörige Attribut („Druckermodell“) aus der Datenbanktabelle „Produkt\_Attribute“ geladen. Jedes Produkt verfügt über ein anderes Druckermodell, wodurch in den Produktbeschreibungen aller  $m$  Produkte (im Prototyp rund 30'000, das Quellprodukt ausgenommen) nach dem Attribut „Druckermodell“ gesucht wird.

Dadurch ergeben sich insgesamt  $n$  simple Datenabfragen für das jeweilige Attribut und  $n$  aufwändige Datenbankabfragen zur Suche nach Produkten, die das Attribut im Produkttext enthalten. Zu jedem Produkt kann eine Vielzahl an Zubehör gefunden werden. Pro gefundenem Zubehörprodukt wird wiederum eine INSERT-Anfrage benötigt, um die Zuordnung Produkt – Zubehör in die Datenbanktabelle „Produkt\_Supplies“ einzufügen.

## Zubehör Zuordnungen berechnen

[Bestehende Zuordnungen löschen](#)

Regel 1:	<input type="button" value="Regel 1 berechnen"/>
Regel 2:	<input type="button" value="Regel 2 berechnen"/>
Regel 3:	<input type="button" value="Regel 3 berechnen"/>
Regel 4:	<input type="button" value="Regel 4 berechnen"/>
Regel ALLE:	<input type="button" value="Regel ALLE berechnen"/>

Abbildung 22

### Zubehör-Berechnungsinterface

(Quelle: [www.electronics3000.ch](http://www.electronics3000.ch), Administrationsbereich, 2010)

Die Zubehöruordnungen werden vorberechnet (Abbildung 22). Dies kann automatisiert geschehen durch einen „Cronjob“ [Linux Glossar 2010], der jeden Arbeitstag am Morgen automatisch ausgeführt wird, oder manuell durch einen Klick auf einen der Berechnen-Buttons.

Die beiden Anforderungen Aktualität und Performance können dadurch unter einen Hut gebracht: Die Performance wird optimiert, da die Zubehörprodukte vorberechnet werden. Beim Aufruf eines Produkts im Onlineshop kann das Zubehör (durch die Indizierung der referenzierten Produkt ID) sehr schnell und einfach aus der Datenbank geladen werden. Die Aktualität ist ebenfalls optimal gewährleistet, da die Produktlisten nur einmal täglich durch die Lieferanten aktualisiert werden. Mit Ausnahme der Lagerbestände, die stündlich auf den neusten Stand gebracht werden. Um zu verhindern, dass ein Kunde ein Lagerprodukt zusammen mit einem nicht verfügbaren Zubehörprodukt bestellt, werden nur Zubehörprodukte angezeigt, deren Lagerbestand positiv ist.

### **5.2.3 Die Bedeutung semantischer Daten**

Ein Zubehörsystem ist auch möglich, das zeigen die Zuordnungsregeln eins bis drei, ohne zusätzliche semantische Daten aus den Produktbeschreibungen zu generieren. Allerdings sind die Möglichkeiten stark eingeschränkt, insbesondere wenn es darum geht, Kompatibilitäten zwischen verschiedenen Produkten herzustellen oder zu ermitteln. Ohne semantische Daten, dazu gehört auch beispielsweise der Sockeltyp eines Prozessors oder die Modellbezeichnung eines Druckers, ist es nicht möglich, zuverlässig passende Zubehörprodukte zuzuordnen. Hinzu kommt, dass sich weitere Zuordnungsregeln in Hinblick auf die Weiterentwicklung von Regeln und Beziehungen zwischen semantischen Daten erdenken lassen. Beispielsweise könnte ein Zubehörsystem die Kompatibilität verschiedener Attribute (z.B. Motherboard-Anschluss und Festplatten-Anschluss, CPU-Sockel etc.) überprüfen, um dadurch einem Konfigurator ähnlich den Kunden bei seinem Kaufentscheid zu unterstützen.

### **5.2.4 Praktischer Nutzen**

Ein Zubehörsystem dient dazu, dem Kunden zusätzliche Produkte anzubieten, um dadurch den Umsatz pro Kunden und den Deckungsbeitrag pro Bestellung zu steigern. Die manuelle Zuordnung von Zubehörprodukten ist eine sehr umständliche Lösung, da täglich einige dutzend bis hundert neue Produkte ins Sortiment gelangen können. Gleichzeitig sind bestehende Zubehörprodukte möglicherweise „End of Life“ und entsprechend nicht mehr lieferbar. Durch eine Mischung aus manuell erstellten Regeln und Vorgaben und einer automatisierten Ausführung und Verwaltung kann das Zubehörsystem effizient umgesetzt werden. Der Wirkungsgrad des Zubehörsystems und der Einfluss auf die Wertschöpfung werden in Kapitel 6.2 ausführlich diskutiert.

### 5.2.5 Ausblick

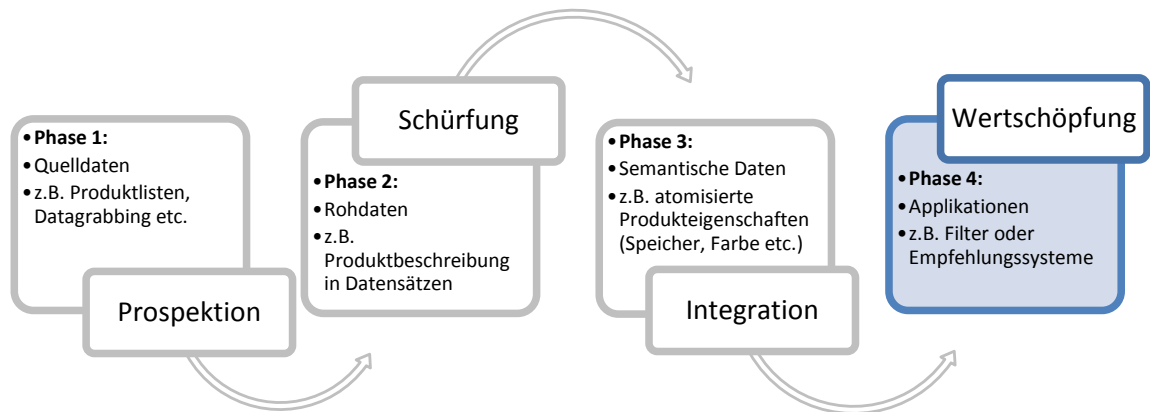
Das Zubehörsystem ist eine Applikation semantischer Daten. Durch weitere Zuordnungsregeln kann dieses System erweitert und verbessert werden. Insbesondere lassen sich die Attribute durch geeignete Regeln derart erweitern, dass dynamisches Zubehör in Form eines Konfigurators angeboten werden kann. So ist es beispielsweise möglich, durch Abgleich der Attribute kompatible Komponenten für einen Computer zu ermitteln und dem Benutzer dadurch eine Alternativauswahl zu geben. Dadurch lassen sich Produkte individualisieren und auf die Bedürfnisse des Kunden anpassen.

Durch das Preisintervall kann die Zubehörzuordnung auf Produkte zwischen einem bestimmten Mindest- und Höchstpreis reduziert werden. Dadurch können allenfalls in der Kategorie integrierte Zubehörprodukte oder Produkte in gewissen Preisschichten aussortiert oder gesondert behandelt werden. Beispielsweise wäre es möglich, für jede Preisklasse einer Produktkategorie unterschiedliche Zubehörprodukte zu definieren. Dadurch kann eine bedürfnisoptimierte Empfehlung erreicht werden, wenn dem anspruchsvollen Kunden zum High-End TV auch ein entsprechend hochwertiges (und teures) Kabel angeboten wird.

In Anlehnung an den Ansatz der „hybriden Empfehlungstechniken“ [Höhfeld/Kwiatkowski 2007] kann die Wirkung nicht-personalisierter Systeme durch Verknüpfung mit kooperativen Methoden und der damit einhergehenden Einbeziehung von Nutzerdaten gesteigert werden. Sobald das Kaltstartproblem überwunden und genug Nutzerdaten vorhanden sind, kann das Zubehörsystem um eine Komponente der „kollektiven Intelligenz“ [Segaran 2008] erweitert werden. Zubehör kann dann nicht nur aufgrund von statischen Regeln generiert, sondern auch aufgrund von Nutzererfahrungen und Nutzerverhalten berechnet und angeboten werden. Ein Popularitätswert könnte beispielsweise nützlich sein, um die Reihenfolge der Auflistung festzulegen.

Bereits die bestehende Version ermöglicht aber eine funktionierende und nützliche Generierung von Zubehörprodukten und unterstützt damit das Ziel, möglichst jedem Kunden ein passendes Zubehörprodukt zu verkaufen. Dadurch steigt unter der Voraussetzung des praktischen Nutzens durch den Kunden der Deckungsbeitrag wie in Kapitel 5.2.4 dargelegt erheblich.

## 6 Wertschöpfung der Integration



**Abbildung 23** Wertschöpfung im formalisierten Prozess

Die Entwicklung eines Systems, das den vollständigen vierstufigen Prozess der semantischen Daten inklusive der Applikationen implementiert, kann einige Wochen beanspruchen und damit schnell Kosten im fünfstelligen Bereich verursachen. Die Investitionen sollen aber Rendite zeigen und betriebswirtschaftliche Kennzahlen des Onlineshops verbessern. Die Wirkung der semantischen Daten als zusätzlich beschreibendes Element in der Produktdarstellung ist dabei kaum messbar. Der Nutzen der implementierten Anwendungen Filtersystem und Zuhörersystem lässt sich aber gut quantifizieren. In dieser Beziehung interessant sind die Umsatz- und Bestellentwicklung sowie die Messdaten, die Erkenntnisse darüber liefern, auf welcher Seite oder in welcher Situation der Kunde nicht bestellt hat.

Im Shopsystem sind Funktionalitäten eingebaut, die den Benutzer anhand der Session und der IP-Adresse identifizieren und seine Interaktion speichern. Dabei wird der Seiten- und Warenkorbverlauf erfasst.

Der betriebswirtschaftliche Nutzen und die Messdaten sollen in Kapitel 6.1 und 6.2 für das Filter- und Zuhörersystem gesondert analysiert und in Kapitel 6.3 in einen übergeordneten betriebswirtschaftlichen Kontext gebracht werden.

## 6.1 Analyse Filtersystem

Das Filtersystem unterstützt den Benutzer in der Produktaufistung durch die Anzeige von verschiedenen Produktattributen, nach denen die Auflistung eingegrenzt werden kann.

Der Nutzen des Filtersystems lässt sich anhand quantitativer und qualitativer Kennzahlen messen. Von besonderem Interesse ist hierbei die Datenbanktabelle „Filtertemp“, die alle Interaktionen des Kunden speichert. Dies ist aus technischer Sicht notwendig, damit gesetzte Filter beim Seitenwechsel (z.B. Klick auf ein Produkt und Anzeige der Produktseite) gespeichert und bei der Rückkehr zur Produktaufistung wiederhergestellt werden können. Pro Produktkategorie wird höchstens ein Datensatz angelegt, und auch nur dann, wenn der Kunde das Filtersystem verwendet. Als Nebeneffekt ergibt sich, dass damit das Kundenverhalten analysiert werden kann.

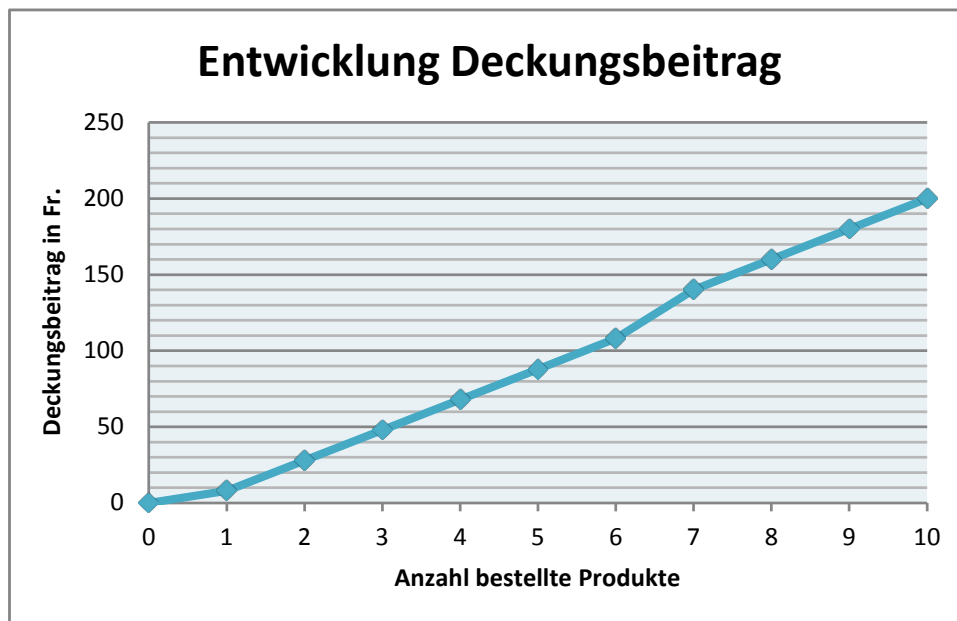
Das Filtersystem ist seit März 2010 in 29 wichtigen Produktkategorien im Einsatz. Dabei wurden 1650 Benutzungen von 1337 Benutzern gespeichert. Insgesamt waren im gleichen Zeitintervall ungefähr 182'440 Benutzer im Onlineshop, rund 30'658 haben dabei auf die Produktaufistung zugegriffen, 15'956 auf Kategorien mit Filtern. Rund 10% haben also die Filter in Anspruch genommen. Berücksichtigend, dass ungefähr 30-40% aller Seitenbesuche auf Suchmaschinen zurückzuführen sind, bedeutet dies also eine Quote von rund einem Drittel bis zur Hälfte aller Benutzer. Interessant ist insbesondere, dass rund 251 Kunden vor ihrer Bestellung das Filtersystem benutzt haben. Das bedeutet, dass 20% aller Filterbenutzer zum Bestellabschluss gelangt sind.

Die Beurteilung der effektiven Wertschöpfung des Filtersystems ist aufgrund der vorhandenen statistischen Daten schwierig, da nicht festgestellt werden kann, wie viele Kunden aufgrund des Filtersystems zur Kaufentscheidung fanden und wie viele Kunden sowieso bestellt hätten. Das Filtersystem bewirkt eine eher indirekte Wertschöpfung, in dem es den Kunden bei seiner Produktsuche und Kaufentscheidung unterstützt. Als approximative Annahme kann der empirische Konversionswert  $\frac{\text{Filterbenutzungen}}{\text{Bestellungen}}$  von rund 20% verwendet werden.

## 6.2 Analyse Zubehörsystem

Das Zubehörsystem unterstützt den Kunden beim Kauf, empfiehlt passendes Zubehör und kann den Umsatz pro Kunden erhöhen. Tatsächlich generiert jedes zusätzlich verkaufte Produkt einen zusätzlichen Deckungsbeitrag, und zwar überproportional zum Deckungsbeitrag des ersten Produkts.

Abbildung 24 zeigt den Verlauf des kumulierten Deckungsbeitrags anhand der Anzahl Produkte, die ein Kunde in derselben Bestellung bestellt. Die Umsatzentwicklung soll an dieser Stelle vernachlässigt werden. Der Deckungsbeitrag beim ersten Produkt, das gleichzeitig das teuerste Produkt ist, beträgt 8 Franken. Durch Hinzufügen des zweiten Produkts resultiert ein zusätzlicher Deckungsbeitrag von 20 Franken.



**Abbildung 24** Entwicklung Deckungsbeitrag

Das zweite Produkt, das um einen Faktor 2.5 günstiger ist als das erste, steigert den kumulierten Deckungsbeitrag also um einen Faktor 3.5. Jedes weitere zusätzliche Produkt trägt weitere 20 Franken bei. Das achte Produkt steigert den Deckungsbeitrag um 32 Franken. Dies hängt damit zusammen, dass ab einem bestimmten Umsatz pro Bestellung die Lieferung durch den Distributor versandkostenfrei erfolgt und die einkalkulierten Versandkosten des ersten Produkts als Deckungsbeitrag verfügbar werden. Der kumulierte Deckungsbeitrag kann durch die lineare Gleichung

$$f(x) = \begin{cases} 20(x - 1) + 8, & x < 7 \\ 20(x - 1) + 20, & x \geq 7 \end{cases}$$

im Speziellen oder durch die allgemeine Form

$$f(x) = \begin{cases} u(x - 1) + t, & x < x_{vk} \\ u(x - 1) + u, & x \geq x_{vk} \end{cases}$$

ausgedrückt werden, wobei

$x = \text{Anzahl bestellte Produkte},$

$t = \text{Marge in Fr.},$

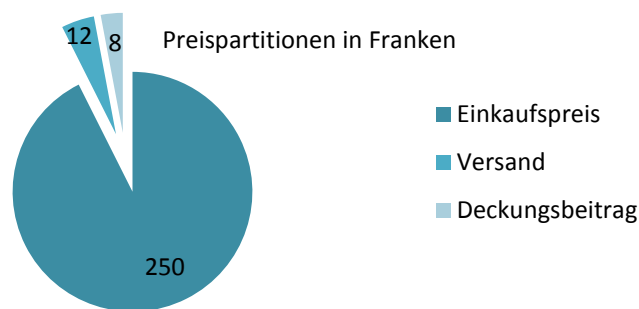
$$u = t + \text{Versandkosten},$$

$$x_{vk} = \text{Versandkostengrenze}$$

Die erste Ableitung der Gleichung liefert den Grenzdeckungsbeitrag  $f'(x) = u$ , wonach also durch jedes zusätzliche Produkt als Deckungsbeitrag die Summe aus konstanter Marge und den konstanten Versandkosten resultiert.

Ein einzelnes Produkt lässt eine Bestellung allenfalls kostendeckend ausfallen. Die wirklichen „Cash Cows“ sind Bestellungen mit mehr als einem Produkt. Insbesondere beim zweiten Produkt (oder dem ersten Zubehör) steigt der erwirtschaftete Deckungsbeitrag deutlich an. Dabei ist diese Entwicklung praktisch unabhängig vom Preis der Produkte, was mit der Einkalkulierung der pauschalen Versandkosten in den Preis jedes einzelnen Produkts und der annähernd konstanten absoluten Marge zusammen hängt.

## Preiszusammensetzung Produkt



**Abbildung 25** Preiszusammensetzung Produkt

Abbildung 25 zeigt die Preiszusammensetzung eines Produkts mit dem Gesamtpreis von 268 Franken, einem Einkaufspreis von 250 Franken, Versandkosten von 12 Franken und einer Bruttomarge von 8 Franken. Dabei sind die Marge und die Versandkosten grundsätzlich über das gesamte Preisspektrum konstant. Ein Produkt mit einem Preis von 3000 Franken liefert also denselben Deckungsbeitrag wie ein Produkt für 100 Franken.

Dieser Ansatz der Preiskalkulation ist eine Folge der hohen Preistransparenz im Internethandel und der daraus resultierenden Fokussierung auf den Preis als entscheidendes Kaufkriterium [Weiber 2002]. Der Preis kann aber auch als Erfolgsfaktor verstanden werden, wenn die Prozesse innerhalb des Unternehmens optimiert sind und dadurch Kostenvorteile generiert werden können, die marktspezifisch für den Erfolg massgeblich sind [Grünig 2009]. Durch Preisvergleichsportale ist es einem Onlineshop mit relativ geringem techni-

schem und finanziellem Aufwand möglich, sehr effiziente Werbung zu betreiben, deren Erfolg vor allem vom Angebot, der Verfügbarkeit und dem Preis abhängt. Damit die variablen Kosten durch die erwirtschafteten Deckungsbeiträge finanziert werden können, sind die Werbe- und Administrationskosten zu berücksichtigen. Die Prozesse der Beschaffung, Distribution, Zahlungs- und Bestellungsverarbeitung sind im Optimalfall bis auf den Kontrollblick eines Mitarbeiters automatisiert. Daraus folgt, dass für ein teures Produkt der gleiche Aufwand besteht wie für ein günstiges Produkt, wodurch sich die pauschale Margenberechnung als vereinfachendes Modell anbietet. Durch Preisdifferenzierungsmechanismen kann die Preis- und Margenberechnung optimiert werden. Ein Ansatz besteht in Preiserhöhungen auf Produkten in weniger umkämpften Märkten. Eine Anpassung der Preise kann insbesondere sinnvoll sein, wenn ein Produkt in Preisvergleichsportalen mit Abstand auf der besten Position rangiert. Dadurch wird Marge verschenkt. Andererseits verursacht jeder manuelle Eingriff in die Preisberechnung und die Überwachung der Preise wiederum einen zusätzlichen administrativen Aufwand, den es zu berücksichtigen gilt.

Das Geschäftsmodell mit der pauschalen Margenberechnung führt zu relativ betrachtet kleinen prozentualen Margen auf höheren Umsätzen, aber einem absoluten tatsächlichen Deckungsbeitrag pro Bestellung. Dieses Modell kann gerade bei Unternehmen, die neu auf dem Markt sind, sinnvoll sein, um in kurzer Zeit ein bestimmtes Umsatzvolumen zu generieren, welches Vorteile beim Einkauf, der Distribution und dem Image herstellt und eine gewisse Planungssicherheit bietet. In diesem Preismodell werden auch die Versandkosten in jedes Produkt eingerechnet. Dadurch verteuert sich eine Bestellung mit mehreren Produkten, da die Versandkosten implizit doppelt verrechnet werden, wodurch der kumulierte Deckungsbeitrag (vgl. Abbildung 24) steigt. Die implizite Mehrfachverrechnung der Versandkosten ist über Preisvergleichsportale nicht direkt ersichtlich (da meistens nur einzelne Produkte verglichen werden können) und damit eine gute Möglichkeit, dem Kunden einerseits kostenlose Lieferung anzubieten, und andererseits die Margen mit jedem zusätzlichen Produkt überproportional zur Marge des ersten Produkts zu steigern.

Es ist also wesentlich, ein Zubehörsystem zu entwickeln, welches dem Kunden mindestens ein zusätzliches Produkt verkaufen kann. Der Preis der Gesamtbestellung oder des Zubehörprodukts ist allerdings für den Deckungsbeitrag praktisch unbedeutend (ab einem bestimmten Umsatz kann die Direktlieferung durch den Lieferant versandkostenfrei erfolgen). Ein Händler verdient an einem teureren Produkt unter Berücksichtigung des diskutierten Preismodells nicht mehr als an einem günstigeren Produkt.

Die Erfahrung mit dem Onlineshop Electronics3000 zeigt, dass der Umsatz einer einzigen Bestellung irrelevant für den Deckungsbeitrag ist. Der Markt ist durch Preissuchmaschinen derart auf den Preis als Erfolgsfaktor fokussiert, dass versandkostenfreie Lieferungen ab einigen tausend Franken Umsatz unmittelbar an den Kunden weitergegeben werden. Ebenso wie mittel- bis langfristige Einsparungen, die durch höhere Einkaufsvolumen bei den Lieferanten und Herstellern erreicht werden können.

Ein Zubehörsystem, das jedem dritten Kunden ein Produkt verkaufen kann, ist also ungefähr gleichbedeutend mit dem Deckungsbeitrag, den sechs Neukunden generieren, die jeweils ein Produkt bestellen.

Die Kosten zur Gewinnung eines Neukunden über Onlinewerbung lassen sich aus der Formel  $K = \frac{\text{Werbekosten in Fr.}}{\text{Anzahl Neukunden}}$  berechnen. Die Werbekosten sind insbesondere abhängig von der Konversionsrate, also dem prozentualen Anteil an geworbenen Nutzern, die tatsächlich als Kunden gewonnen werden können. Auf Basis eines Klickkostenpreises von 10 Rappen und einer Konversionsrate von 2.5% verursacht Onlinewerbung pro Lead Kosten von ungefähr 4 Franken. Der Deckungsbeitrag abzüglich der Marketingkosten beträgt bei einem Kunden mit zwei Produkten also 24 Franken bzw. verallgemeinert

$$DB - K_{Lead} = f(x) - K_{Lead} = u(x - 1) + t - \frac{\text{Werbekosten in Fr.}}{\text{Anzahl Neukunden}}$$

was dem Deckungsbeitragsäquivalent abzüglich der Marketingkosten von sechs Kunden mit einem Produkt entspricht.

$$6 * (f(1) - K_{Lead}) = f(2) - K_{Lead}$$

Eine Bestellung mit zwei Produkten generiert also denselben Deckungsbeitrag wie sechs einzelne Bestellungen.

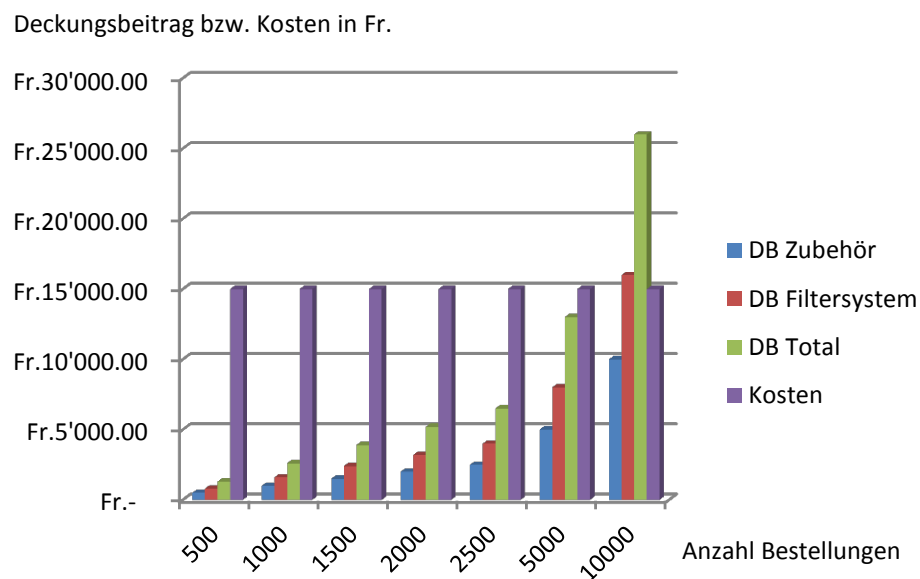
Natürlich ist diese Deckungsbeitragsentwicklung an die Preisberechnung des jeweiligen Onlineshops gebunden. Da aber immer mehr Anbieter die Versandkosten in jedes Produkt einrechnen, um versandkostenfreie Lieferung anzubieten, ist der mit dem zweiten Produkt stark steigende Deckungsbeitrag ein allgemeines Phänomen.

Beim Filtersystem ergibt sich eine Konversionsrate (Conversion Rate) von rund 20%. Tatsächlich liegt dieser Wert höher, wenn die testweise Benutzung und Nutzung zu Beratungszwecken abgezogen wird. Erreicht das Zubehörsystem ebenfalls eine Konversionsrate (d.h. ein zusätzliches Produkt pro Bestellung) in diesem Bereich, können die Marketingausgaben für fünf Kunden gespart werden, ohne den gesamthaften Deckungsbeitrag zu reduzieren.

### 6.3 Kalkulation

Das Ziel jeder üblichen Investition ist auf kurz oder lang die Erzielung von Rendite. Die Entwicklung des Attributsystems und der darauf basierenden Filter- und Zubehörapplikation beansprucht eine tiefe fünfstellige Summe, hinzu kommen Initialaufwand zur Definition der Filter und Zuordnungen sowie ein gewisser Test- und Prüfaufwand.

Kunden, die das Filtersystem benutzen, bestellen in rund 20% aller Fälle. Für das Zubehörsystem sei die Annahme gegeben, dass 5% aller Kunden ein zusätzliches Produkt bestellen und dadurch einen zusätzlichen Deckungsbeitrag von mindestens CHF 20.00 generieren.



**Abbildung 26** Deckungsbeiträge und Kosten

Bei einer Annahme von geschätzten einmaligen Kosten in der Höhe von CHF 15'000.00 und vernachlässigbaren Wartungskosten zeigt sich, dass die Investition bei einem jährlichen Bestellvolumen von 5000 Bestellungen (x-Achse) bereits weitgehend amortisiert wird. Unberücksichtigt der Tatsache, dass sich alle Kunden durch die generierten Produktattribute besser und konkreter informieren können, dass Verkaufsberater sich Suchzeit für Standard-Zubehör sparen können und einige Kunden möglicherweise mehrere Zubehörprodukte bestellen. Bei einem Onlineshop mit 2500 Bestellungen pro Jahr stellt sich die Amortisierung nach zwei Jahren ein. Bei 5000 oder sogar 10000 Bestellungen stellt sich innerhalb von ein bis zwei Jahren eine hohe Rendite ein. Zu beachten ist insbesondere, dass die Kosten praktisch unabhängig von der Anzahl Kunden sind. Desto mehr Bestellungen generiert werden, desto mehr verteilen sich die investierten Kosten und desto schneller kann Rendite

erwirtschaftet werden. Die Grenzverwaltungskosten jeder zusätzlichen Bestellung sind minim, da praktisch der gesamte Prozess automatisiert abläuft.

Durch diese Fixkosten bietet es sich an, die entwickelten Lösungen in neuen Anwendungsgebieten einzusetzen. Im Hinblick auf die Ausweitung des Zubehörsystems auf personalisierte Empfehlungen ist ein Newsletter mit individualisiertem Angebot (basierend auf einer Kombination aus Zubehörsystem und persönlichen Empfehlungen aufgrund der bisher getätigten Bestellungen) eine zu prüfende Möglichkeit. Der zusätzlich anfallende Investitionsaufwand beschränkt sich auf die Analyse der bisherigen Bestellungen und der Integration des bestehenden Zubehörsystems als Empfehlungsmethode. Die zu erwartende Wertschöpfung ist bedingt durch die wiederkehrende Notwendigkeit von Verbrauchsgütern wie beispielsweise passenden Druckerpatronen beträchtlich.

Um die aus den Investitionen erwachsende Wertschöpfung zu optimieren, erscheint die Betrachtung der Applikationen als in übergeordnete Systeme integrierbare Module sinnvoll und nützlich. Dadurch eröffnen sich neue Geschäftsfelder und Möglichkeiten.

#### **6.4 Weitere Geschäftsmodelle und Strategien**

Mit der Entwicklung einer Individualsoftware, die semantische Daten generiert, ist eine Grundlage geschaffen, um die Wertschöpfung durch Applikationen dieser Daten zu steigern. Abhängig vom Bestellvolumen, der Umsatzstärke eines Unternehmens und der Marktsituation kann es sinnvoll sein, weitere Geschäftsmodelle in Betracht zu ziehen, um die Fixkosten der Entwicklung und die damit verbundenen Risiken zu diversifizieren.

Individualsoftware ist „Software, die für den Einsatz in einem speziellen Betrieb entwickelt wird, entweder durch Eigenerstellung (Softwarelebenszyklus) oder Auftragsvergabe an externen Softwarehersteller.“ [Gabler 2010]. Falls die Entwicklung durch Eigenerstellung erfolgt ist, verfügt das Unternehmen über sämtliche Rechte, die vollständige Kontrolle und die Flexibilität, den Quellcode auf spezifische Anforderungen anzupassen. Es bietet sich an, die Individualsoftware zu modularisieren und in einem zusätzlichen Geschäftsfeld als Standardsoftware zu etablieren. Insbesondere wenn der Grund zur Entwicklung einer Individualsoftware darin liegt, dass es „keine bekannte, geeignete Standardsoftware“ [Wikipedia 2010] auf dem Markt gibt. Durch die Bearbeitung der Geschäftsfelder „Onlineshopping“ und „Software für Shopsysteme“ ergeben sich eine Zweigleisigkeit und damit eine Verteilung des Risikos und der Fixkosten auf mehrere Grundpfeiler des Unternehmens. Durch die Synergieeffekte bzw. Querfinanzierung der Individualsoftware (Ertrag durch interne Integration und externen Vertrieb) können Kostenstrukturen optimiert und Weiterentwick-

lungen querfinanziert werden. Durch die Integration von Modulen der eigenen Softwarelösung in externe Systeme kann zudem als Kollateraleffekt Erfahrung, Resonanz und Feedback eingeholt werden, das zur allgemeinen Verbesserung des Systems beiträgt.

Damit mit der als Individualsoftware entwickelten Lösung auch die Bedürfnisse der Drittkunden befriedigt werden können, bietet sich die Modularisierung nach Phasen an. Die Prospektion und die Schürfung sind sehr generisch, aber auf einen bestimmten Schnittstellentyp entwickelt. In der Phase der Prospektion können beliebige Daten in eine relationale Datenbank geladen werden, wobei einzig die Quelldatei einer bestimmten Anforderung hinsichtlich des Formats genügen muss. Die Zuordnung von Quellspalten und Datenbankspalten kann über das in Kapitel 3.1 präsentierte Interface erfolgen und ist damit sehr flexibel einsetzbar. In der Phase der Schürfung können beliebige Daten aus einer relationalen Datenbank geladen und auf Basis regulärer Ausdrücke Attribute generiert werden. Hierbei kann es sich um beliebige Attribute handeln, die einzig der Bedingung der Berechenbarkeit durch reguläre Ausdrücke genügen müssen. Aus Prospektion und Schürfung lassen sich also zwei Module formen, die sich als Elemente des Geschäftsfelds „Software für Shopsysteme“ anbieten.

Eher weniger geeignet für einen Vertrieb sind die konkreten Integrationen der Phase 3. Filter- und Empfehlungssysteme sind stark in das Shopsystem eingebunden und auf dessen Struktur abgestimmt. Anstelle eines vordefinierten Moduls ist es denkbar, die Integration als Individuallösung zu verkaufen, die auf den standardisierten Modulen 1 und 2 aufbaut. Dadurch können die Kosten gering, die Wiederverwendbarkeit von Quellcode und die Flexibilität hoch gehalten werden.

Das Geschäftsfeld „Software für Shopsysteme“ muss explizit als Nischenprodukt konzipiert und verkauft werden. Die Module sind optimal auf den einheimischen Schweizer Markt abgestimmt, insbesondere die Schnittstellen auf die standardisierten Hersteller- und Lieferantenlisten ausgelegt. Unternehmen aus der IT-/CE Branche könnten die Module praktisch unverändert übernehmen, da die Lieferanten und damit die Schnittstellenkonfigurationen dieselben sind. Aufgrund der starken Individualisierung der Software auf den Inlandmarkt Schweiz kann der Gefahr von Offshoring [Deutsche Bank 2010] vorgebeugt werden. Andererseits entsteht die Gefahr eines Verdrängungs- oder Kannibalisierungseffekts im Geschäftsbereich „Onlineshopping“ durch den Geschäftsbereich „Software für Shopsysteme“, da diese zu Verbesserungen der Shopsysteme der Konkurrenz führen. Entsprechend ist ein wesentlicher Erfolgsfaktor der Umsatzzuwachs des neuen Geschäftsbereichs, der grösser sein muss als der Umsatz, der dadurch im bestehenden Geschäftsbereich

verloren geht [Schögel 2009]. Alternativ bietet sich an, die Module primär an Unternehmen nicht konkurrierender Branchen zu vertreiben. Aufgrund der generischen Entwicklung und der Schnittstellen ist dies mit relativ kleinem Aufwand zu bewerkstelligen, zudem wird das Risiko der Selbstkonkurrenzierung über die Unterstützung von Konkurrenten minimiert.

### 6.5 Substitutionsanalyse

Grundsätzlich muss keine Software entwickelt werden, um allgemeine Produktattribute zu generieren. Die entsprechenden Daten können bei Informationsdienstleistern eingekauft werden (z.B. C-NET Content Solutions), wodurch eine Software zur Datenschürfung substituiert bzw. überflüssig wird. Dabei fallen Kosten von ungefähr CHF 780.00 monatlich bei 20'000 Datensätzen an [Proseller 2010]. Auf ein Jahr berechnet ergibt sich ein Nettoaufwand von CHF 9'360.00, was zzgl. MwSt. und Integrationskosten ungefähr deckungsgleich mit dem in Kapitel 6.3 veranschlagten Investitionsbetrags zur Entwicklung eines semantischen Informationsverarbeitungssystems ist. Beim Dateneinkauf handelt es sich allerdings um wiederkehrende Fixkosten, die während der gesamten Dauer der Geschäftstätigkeit verrechnet werden. Ebenso handelt es sich nur um semantische Daten, die eingekauft werden. Die Integrationsmodule und Anwendungen müssen systemspezifisch zusätzlich entwickelt werden.

	<b>Individualsoftware</b>	<b>Dateneinkauf</b>
<b>Datensätze</b>	unbegrenzt	20'000
<b>Einmalige Kosten</b>	ca. CHF 15'000	ca. CHF 5'000
<b>Laufende Kosten</b>	Wartungskosten	ca. CHF 780 monatlich zzgl. Wartungskosten
<b>Datenqualität</b>	Benutzer- und Listenabhängig	Quellenabhängig
<b>Datentiefe</b>	Abhängig Quellen	durch Recherchen hoch
<b>Synergien mit anderen Geschäftsfeldern</b>	Module können vertrieben werden	Keine

Tabelle 2

Individualsoftware und Dateneinkauf

Tabelle 2 vergleicht die wichtigsten Aspekte der Individualsoftwarelösung und des Dateneinkaufs zur Integration eines semantischen Informationssystems in eine bestehende Softwarelösung (z.B. Onlineshop).

Obwohl die einmaligen Kosten bei der Individualsoftware wesentlich höher sind als beim Dateneinkauf, amortisieren sich diese Investitionen bereits nach knapp einem Jahr wieder. Falls zusätzliche Geschäftsfelder erschlossen und einzelne Module der Entwicklung vertrieben werden, kann die Wertschöpfung zusätzlich gesteigert und die Amortisation unterstützt werden.

Der Dateneinkauf ist vor allem eine gute Lösung, falls detaillierte Produktinformationen in ein bestehendes System integriert werden sollen, wobei der Produktkatalog relativ statisch und auf populäre Produkte ausgerichtet ist (die in der Datenbank des Datenlieferanten erfasst sind). Zudem sollte das Unternehmen bereits einen gewissen Bekanntheitsgrad haben, damit sich die relativ hohen wiederkehrenden Kosten rechnen.

Die Individualsoftware ist eine gute Lösung, falls bestehende Produktinformationen aus Listen schnell, aktuell und zeitnah nach den wichtigsten Attributen verarbeitet werden sollen. Aufgrund der Flexibilität des Systems fallen kaum Wartungskosten an, die Regeln zur Generierung semantischer Daten bzw. die regulären Ausdrücke müssen nur bei komplett neuen Kategorien oder Änderungen der Beschreibungsstruktur anwenderseitig angepasst werden. Gerade für Unternehmen mit Nischenprodukten ist diese Lösung zu bevorzugen, da Informationsdienstleister oftmals mehrheitlich Daten zu populären Produkten anbieten. Mit der Softwarelösung hat das Unternehmen aber die Flexibilität, aus jeder Produktbeschreibung Attribute zu generieren. Ein Produkt ohne Beschreibung existiert nicht, ansonsten hätte kein Kunde die Möglichkeit, zu identifizieren, um was für ein Produkt es sich handelt. Es kann höchstens tiefere und oberflächlichere Beschreibungen geben, allerdings kann dies auch beim Dateneinkauf der Fall sein.

Einem grösseren Unternehmen, das die populärsten und bekanntesten Produkte anbietet und dabei auf Produktbeschreibungen angewiesen ist, kann der Dateneinkauf sicherlich empfohlen werden. Für Unternehmen mit kleineren Kapitalien und allenfalls einem Angebot an Nischen- oder lokalen Produkten, ist die Datengenerierung mittels Software aufgrund der ausgeführten Argumente zu bevorzugen. Rein betriebswirtschaftlich betrachtet verursacht der Dateneinkauf bereits nach einem Jahr die höheren kumulierten Kosten als die Softwarelösung.

Zur abschliessenden Beurteilung von Dateneinkauf oder Softwarelösung müsste eine Studie die Wirkung der entsprechenden Daten auf die Kunden und das Bestellverhalten analysieren. Falls die höhere Datentiefe des Dateneinkaufs die höheren Kosten nicht aufwertet, wäre aus betriebswirtschaftlicher Sicht ebenfalls die Softwarelösung zu bevorzugen. Falls sich aber zeigt, dass die allenfalls tieferen Informationen des Dateneinkaufs einen höheren Anteil der Kunden zum Kauf bewegen, könnten die höheren Kosten bei entsprechendem Bestellvolumen kompensiert werden.

Eine Studie zeigt, dass Kunden sich verstärkt über spezialisierte Testberichts-, Verbrauchs- und Preisvergleichsportale über das gewünschte Produkt informieren [Bitkom 2010]. Dadurch hat ein Kunde den Kaufentscheid bereits auf ein bestimmtes Produkt fokussiert, bevor er den Onlineshop betritt. Durch eine umfassende, wenn auch nicht tiefere, Produktbeschreibung im Onlineshop des Unternehmens kann das Vertrauen des Kunden erweckt werden, auch wirklich das gewünschte Produkt zu bestellen (was nicht unbedingt gegeben ist, wenn beispielsweise nur eine Modellnummer angezeigt wird). Aufbauend auf dieser Annahme wäre die Kombination aus automatisch generierten Produktattributen und Herstellerlisten also ausreichend, um das Informationsbedürfnis des Kunden zu befriedigen und damit die aus einem Dateneinkauf resultierenden tieferen Produktbeschreibungen nicht entscheidungsrelevant (und damit betriebswirtschaftlich betrachtet überflüssig).

## 7 Schlusswort

### 7.1 Zusammenfassung

Innerhalb dieser Forschungsarbeit wurde aufgezeigt, wie die Generierung und Integration semantischer Daten in einem vierstufigen Prozess formalisiert und implementiert werden kann. Die Datenschürfung mittels regulärer Ausdrücke hat sich als besonders flexibel und zuverlässig erwiesen, um Produktattribute zu generieren und in einer relationalen Datenbank zu speichern. Diese Produktattribute werden als semantische Daten bezeichnet, als die kleinste bedeutsame Informationseinheit.

Zur Integration der semantischen Daten in einen Onlineshop wurden ein Filter- und ein Zubehörempfehlungssystem entwickelt und diskutiert. Das Filtersystem unterstützt den Benutzer bei der Produktsuche, in dem es die Produktauflistung auf die Produkte beschränkt, die bestimmten, ausgewählten Eigenschaften genügen. Diese Filterung basiert auf den Produktattributen, die in der Datenschürfung generiert worden sind. Das Zubehörsystem ist konzipiert als nicht-personalisiertes Empfehlungssystem. Es generiert aus vier verschiedenen Regeltypen zu beliebigen Produkten kompatibles Zubehör und speichert die entsprechenden Beziehungen in der Datenbank. Aufgrund der in die Preise einkalkulierten Versandkosten kommt dem Verkauf von Zubehörprodukten in Bezug zum erzielten Deckungsbeitrag eine hohe Bedeutung zu.

Durch das Filter- und Zubehörsystem wird eine Steigerung der Wertschöpfung erzielt, wie aus empirischen Betrachtungen der Nutzerdaten und einer argumentativen Analyse des Preismodells hervor geht. Die angefallenen Investitionen können durch eine Modularisierung des vierstufigen Prozesses und dem Vertrieb dieser Module in einem neuen Geschäftsfeld schneller amortisiert werden. Eine Alternative zur Generierung semantischer Daten besteht im Einkauf der Informationen. Diese Vorgehensweise rechnet sich jedoch nur bei grösseren Unternehmen mit Fokus auf den breiten Markt und allenfalls bereits bestehender Integrationsanwendungen.

### 7.2 Ausblick

Aufgrund der hohen Bedeutung von Zubehörprodukten bzw. zusätzlichen Produkten in der Bestellung liegt eine Weiterentwicklung des Zubehörsystems nahe. Durch die Einbindung von personalisierten Verfahrensweisen ist es denkbar, das Zubehörangebot zu optimieren und auf die individuellen Kunden und Warenkörbe anzupassen. Eine Kombination aus nicht-personalisiertem und personalisiertem Empfehlungssystem erlaubt die Vorteile von automatischer Berechnung und kollektiver Intelligenz zu kombinieren und dadurch die

Wertschöpfung zu steigern. Diese hybride Lösung führt das Zubehörsystem über ein regelbasiertes, deterministisches System hinaus zu einem bedürfnisspezifisch, individuellem System. Dieses setzt allerdings gewisse Datenbestände voraus, um zuverlässig und effektiv funktionieren zu können.

Ein weiterer Ansatz, die semantischen Daten zu integrieren, besteht in der Entwicklung eines Produktkonfigurators. Nach dem Filter- und dem Zubehörsystem beschreitet dieser eine höhere Stufe der Analyse der Beziehungen semantischer Daten, damit die Kompatibilität der Konfigurationen sichergestellt wird. Gleichzeitig bedeutet dies einen weiteren Schritt in Richtung Personalisierung von Applikationen, in dem Produkte individualisiert werden können.

Semantische Daten dürften in den nächsten und Jahrzehnten an Bedeutung gewinnen, wenn im Sinne des „Web 3.0“ das Internet von der reinen Informationspräsentation zur semantischen Auszeichnung übergeht, damit Informationssysteme die Bedeutung von Daten verstehen können.

Kurz- bis mittelfristig wird die Integration der entwickelten Applikationen im Onlineshop Electronics3000.ch zeigen, inwiefern die betriebswirtschaftlichen Zahlen verbessert und beispielsweise durch das Zubehörsystem Kunden zu zusätzlichen Käufen animiert werden können. Ebenso ist davon auszugehen, dass sich weitere Anwendungen zur Nutzung der semantischen Datenbestände erschliessen, die das Einkaufserlebnis für den Kunden und die Wertschöpfung für das Unternehmen weiter steigern.

### **7.3 Kritische Würdigung**

Ein breites Spektrum an Ideen und Konzepten musste auf einen kleinen spezialisierten Bereich konzentriert werden, um diesen intensiv zu recherchieren und zu studieren und damit als Gegenstand dieser Arbeit zu fokussieren. Der Spezialisierung auf die Thematik der semantischen Daten im Kontext eines spezifischen Informationssystems musste das Forschungsobjekt Produktkonfigurator weichen. Ein Konfigurator führt über semantische Daten hinaus tiefer in das Gebiet der Ontologien und komplexen Datenbeziehungen, was den Rahmen dieser Arbeit gesprengt hätte. Auf Basis des formalisierten vierstufigen Prozesses und des entwickelten Systems ist es jedoch in einem nächsten Schritt möglich, einen Konfigurator aufzubauen und zu integrieren.

Das entwickelte System ist flexibel und erlaubt konzeptionell eine bestimmte generische Modularisierung, damit die Komponenten einzeln vertrieben werden können. Die Fertigkeiten zur Schürfung semantischer Daten sind ausgeprägt und erlauben zuverlässige Abläu-

fe. Die Methoden zur Integration stellen einen Fortschritt für das Benutzerhandling des Informationssystems und für die Verkaufsförderung dar. Auch wenn diese gerade im Bereich der Personalisierung durchaus noch Innovationspotential haben, leisten sie im Hinblick auf die innerhalb dieser Arbeit spezifizierten Anforderungen zuverlässige und wertschöpfungsfördernde Dienste.

#### **7.4 Schlusswort**

Innerhalb dieser Arbeit konnte anhand eines konkreten Exempels gezeigt werden, wie Forschung und Theorie der Wirtschaft Hand bieten können, um Prozesse und neue Verfahren zu entwickeln und umzusetzen und die Wertschöpfung bestehender Informationssysteme zu steigern. Damit ist ein Exempel statuiert, dass akademische Forschung und Wirtschaftspraxis keineswegs aneinander vorbeigehen oder unabhängig voneinander existieren müssen. Es ist auch gerade die Faszination, Theorie und Praxis zu vereinen und die Resultate in der realen Wirtschaftswelt zu evaluieren, die Antrieb zu dieser Arbeit waren. Wie im Ausblick (vgl. Kapitel 7.2) dargelegt, gibt es verschiedene Ansätze, die Forschung in diesem Bereich weiterzuführen und das System weiterzuentwickeln. Interessant wird die Wertschöpfungsanalyse des Systems über einen mittelfristigen Zeitraum sein. Hierbei wird sich zeigen, inwiefern die angestrebten Ziele und das Kalkulationsmodell aus Kapitel 6 von der Praxis divergieren.

## Literaturverzeichnis

- [Gabler 2010] Gabler Wirtschaftslexikon: *Individualsoftware*, zugegriffen am 05. November 2010.
- [Schögel 2009] Schögel, Marcus: *Multichannel Management*, verfügbar: [http://www.promarca.ch/uploads/media/PwC\\_HSG\\_PromarcaForum\\_2009.pdf](http://www.promarca.ch/uploads/media/PwC_HSG_PromarcaForum_2009.pdf), zugegriffen am 05. November 2010
- [Proseller 2010] Proseller AG: *Kosten C-NET*, verfügbar: <http://www.concertopro.ch/concerto-data/produkte-und-preise-2.htm>, zugegriffen am 05. November 2010
- [Bitkom 2010] Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V., *Mehrheit informiert sich vor einem Kauf im Web*, verfügbar: [http://www.bitkom.org/de/presse/8477\\_62717.aspx](http://www.bitkom.org/de/presse/8477_62717.aspx), zugegriffen am 05. November 2010
- [Grünig/Kühn 2009] Grünig Rudolf; Kühn, Richard: *Methodik der strategischen Planung*, 5. Auflage, Haupt, Stuttgart, 2009.
- [Stormer 2010] Stormer, Henrik: *Personalisierung in Online Shop Systemen: Produktkonfiguration, Empfehlung Adoption*, Dissertation, Universität Fribourg, 2008.
- [Meier 2007] Meier, Andreas: *Relationale und postrelationale Datenbanken*, 6. Auflage, Springer, Berlin, 2007.
- [Sury 2007] Sury, Ursula: Content Grabbing – Was tun? *Informatik Spektrum*, Jg. 07, Heft 6 (Juni 2007), S.457
- [Meier/Stormer 2010] Meier, Andreas; Stormer, Henrik: *eBusiness & eCommerce: Management der digitalen Wertschöpfungskette*, 2. Auflage, Springer, Berlin, 2008.
- [Gräfe 2005] Gräfe, Gernot: *Informationsqualität bei Transaktionen im Internet*, 1. Auflage, Gabler, Wiesbaden, 2005, S. 50.
- [Deutsche Bank 2004] Deutsche Bank Research: *Offshoring: Globalisierungswelle erfasst Dienstleistungen*, verfügbar: [http://www.dbresearch.de/PROD/DBR\\_INTERNET\\_DE-PROD/PROD000000000178654.pdf](http://www.dbresearch.de/PROD/DBR_INTERNET_DE-PROD/PROD000000000178654.pdf), zugegriffen am 05. November 2010
- [Back/Gronau/Tochtermann 2008] Back, Andrea; Gronau, Norbert; Tochtermann, Klaus: *Web 2.0 in der Unternehmenspraxis*, 1. Auflage, Oldenbourg, 2008.
- [HMD 271, 2010] Meier, Andreas; Hengartner, Urs: *Web 3.0 & Semantic Web. Praxis der Wirtschaftsinformatik*, Heft 271 (Februar 2010).
- [Weinhofer 2010] Weinhofer, Joachim: *Extraktion semantisch relevanter Daten aus natürlich sprachlichen Inhalten in Hinblick auf eine automatische Fragengenerierung*, Masterarbeit, Technische Universität Graz, 2010.
- [Schukat-Talamazzini 2010] Schukat-Talamazzini: *Mustererkennung*, verfügbar: <http://www.minet.uni-jena.de/fakultaet/schukat/ME/Scriptum/lect-intro.pdf>, zugegriffen am 05. November 2010
- [Pohl/Litfin/Weber 2002] Pohl, Alexander ; Litfin, Thorsten ; Weber, André : *Handbuch Electronic Business : Informationstechnologien – Electronic Commerce – Geschäftsprozesse : Marketing-Mix-bezogene Gestaltungselemente des Internet-Auftritts*. Gabler, Wiesbaden, 2002, S. 655-669

- [Wikipedia 2010] Wikipedia : *Individualsoftware*, verfügbar : <http://de.wikipedia.org/wiki/Individualsoftware>, zugegriffen am 05. November 2010
- [Linux Glossar 2010] Linux Glossar : *Cronjob*, verfügbar : <http://www.linux-ag.com/fachinformationen/glossar/>, zugegriffen am 05. November 2010
- [Höhfeld/Kwiatkowski 2007] Höhfeld, Stefanie ; Kwiatkowski, Melanie : Information Wissenschaft & Praxis. *Empfehlungssysteme*, 2007, S. 265-276.
- [Sorge 2008] Sorge, Christoph : *Selbstorganisierende Empfehlungssysteme im Internet*, 1. Auflage, Kit Scientific Publishing, 2008, S. 19-20
- [Zhou 2010] Zhou, Tao : *Statistical mechanics of information systems: information filtering on complex networks*, Dissertation, Universität Fribourg, 2010.
- [Thomson/Welling 2007] Thomson, Laura ; Welling, Luke : *PHP 5 & MySQL 5 : Dynamische Webanwendungen von Einstieg bis E-Commerce*, 3. Auflage, Markt & Technik , München, 2007.
- [Kannengiesser 2007] Kannengiesser, Matthias ; Kannengiesser, Caroline : *PHP5 & MySQL 5*, 2. Auflage, Franzis Professional Series, Poing, 2007.

## Anhang A

Dieser Anhang listet alle Datenbanktabellen auf, die im Zusammenhang mit den in dieser Forschungsarbeit diskutierten Systemen und Implementationen von Bedeutung sind.

### Tabellenstruktur für Tabelle Attribute

Feld	Typ	Null	Standard
<i>ID</i>	int(5)	Ja	NULL
Bezeichnung	varchar(30)	Ja	NULL
Sort	int(1)	Ja	0
GlossarID	int(5)	Ja	0
Ausgabetyyp	int(1)	Ja	NULL

### Tabellenstruktur für Tabelle Filter1

Feld	Typ	Null	Standard
<i>ID</i>	int(10)	Ja	NULL
Kat1	int(5)	Ja	NULL
Kat2	int(5)	Ja	NULL
Kat3	int(5)	Ja	NULL
attributkategorie	int(10)	Ja	NULL
attributstring	varchar(20)	Ja	NULL
suchstring	varchar(20)	Ja	NULL
anzahlzeichen	int(2)	Ja	NULL
numerischesattribut	int(1)	Ja	NULL
vorhernachher	varchar(10)	Ja	NULL
umwandlungsstring	varchar(10)	Ja	0
umwandlungsfaktor	float	Ja	0
umwandlung	int(1)	Ja	NULL
tabelle	varchar(50)	Ja	NULL
spaltenname	varchar(50)	Ja	NULL
tabellenKey	varchar(20)	Ja	NULL
RegExprTabelle	text	Ja	NULL
RegExprSQL	text	Ja	NULL
RegExprPHP	text	Ja	NULL
RegExprPHPSpalten	text	Ja	NULL
RegExprKonk	varchar(100)	Ja	NULL
RegExprKonkAktiv	int(1)	Ja	NULL
RegExprTyp	int(1)	Ja	NULL
RegExprPostfix	varchar(40)	Ja	NULL
RegExprValueNum	int(10)	Ja	NULL
RegExprValueText	varchar(100)	Ja	NULL
RegExprNoEmpty	int(1)	Ja	NULL

RegExprConversionAktiv	int(1)	Ja	NULL
RegExprConversionString	varchar(250)	Ja	NULL
RegExprConversionFactor	varchar(250)	Ja	NULL
filter	varchar(1)	Ja	1
Aktiv	int(2)	Ja	2

### Tabellenstruktur für Tabelle Filtertemp

Feld	Typ	Null	Standard
<b>UserID</b>	varchar(100)	Ja	NULL
<b>KatID</b>	int(10)	Ja	NULL
Aktiviert	text	Ja	NULL
MinPreis	decimal(10,2)	Ja	NULL
MaxPreis	decimal(10,2)	Ja	NULL
Filterstring	text	Ja	NULL
Sortierung	varchar(30)	Ja	NULL
Erstelldatum	timestamp	Ja	CURRENT_TIMESTAMP

### Tabellenstruktur für Tabelle Produkt\_Attribute

Feld	Typ	Null	Standard
<b>ProduktID</b>	int(10)	Ja	NULL
Attribut_Nummerisch	int(10)	Ja	NULL
Attributtext	varchar(30)	Ja	NULL
<b>Kategorie</b>	int(5)	Ja	NULL
Status	int(2)	Ja	NULL
AttributfilterID	int(10)	Ja	NULL

### Tabellenstruktur für Tabelle Supplies

Feld	Typ	Null	Standard
<b>ID</b>	int(10)	Ja	NULL
RegelID	int(3)	Ja	NULL
Zielkat	int(10)	Ja	NULL
Eigenschaft1	int(10)	Ja	NULL
Eigenschaft2	int(10)	Ja	NULL
Eigenschaft3	int(10)	Ja	NULL
Eigenschaft4	int(10)	Ja	NULL
Eigenschaft5	int(10)	Ja	NULL
Eigenschaft6	int(10)	Ja	NULL
Eigenschaft7	int(10)	Ja	NULL
Eigenschaft8	int(10)	Ja	NULL
Eigenschaft9	int(10)	Ja	NULL

Eigenschaft10	int(10)	Ja	NULL
Text1	text	Ja	NULL
Text2	text	Ja	NULL
Text3	varchar(150)	Ja	NULL
Text4	varchar(150)	Ja	NULL
Preis1	decimal(10,2)	Ja	NULL
Preis2	decimal(10,2)	Ja	NULL
TitelID	int(10)	Ja	0
Aktiv	int(1)	Ja	1

### Tabellenstruktur für Tabelle Supplies\_Titel

Feld	Typ	Null	Standard
<b>ID</b>	int(10)	Ja	NULL
DE	varchar(150)	Ja	NULL
FR	varchar(150)	Ja	NULL
ENG	varchar(150)	Ja	NULL

### Tabellenstruktur für Tabelle Zugriffstatistik

Feld	Typ	Null	Standard
<b>ID</b>	int(25)	Ja	NULL
Zeit	timestamp	Ja	CURRENT_TIMESTAMP
SessionID	varchar(40)	Ja	NULL
IP	varchar(20)	Ja	NULL
Sparte	int(3)	Ja	NULL
Kat3	int(5)	Ja	NULL
ProduktID	int(10)	Ja	NULL
Bereich	varchar(15)	Ja	NULL
KundenID	int(20)	Ja	NULL
Quelle	varchar(5)	Ja	-1

**Anhang B**

Die relevanten Programmcodes werden auf dem beiliegenden Datenträger zur Verfügung gestellt.



## ERKLÄERUNG

Ich versichere, dass ich die vorstehende Arbeit selbständig angefertigt und entsprechend den Grundsätzen wissenschaftlicher Ehrlichkeit abgefasst habe.

Es ist mir bekannt, dass andernfalls die Abteilung gemäss dem Fakultätsrat vom 09.11.2004 das Recht hat, den auf Grund dieser Arbeit verliehenen Titel zu entziehen.

....., den .....20.....

.....

(Unterschrift)