Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Andre Helpap

Datenintegration in KMU: Eine Evaluation ausgewählter Tools

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Andre Helpap

Datenintegration in KMU: Eine Evaluation ausgewählter Tools

Abschlussarbeit eingereicht im Rahmen Bachelorarbeit

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Olaf Zukunft
Zweitgutachter: Prof. Dr. Sarstedt

Abgegeben am 10.01.2020

Andre Helpap

Thema der Bachelorarbeit

Datenintegration in KMU: Eine Evaluation ausgewählter Tools

Stichworte

Pentaho, Talend, Open Source, Datenintegration, Business Intelligence, ETL, KMU

Kurzzusammenfassung

Diese Arbeit evaluiert die Open Source Datenintegrationstools von Pentaho und Talend auf ihre Eignung für kleine und mittlere Unternehmen (KMU). Dabei wird auch ein Vergleich zwischen beiden Anwendungen gezogen. Die Evaluation erfolgt exemplarisch an einem spezifischen KMU.

Andre Helpap

Title of the paper

Data Integration in SME: An evaluation of selected tools

Keywords

Pentaho, Talend, Open Source, Data Integration, Business Intelligence, ETL, SME

Abstract

In this paper the open source data integration tools of Pentaho and Talend are evaluated on their suitability for small-to-medium enterprises (SME). It also contains a comparison between both tools. A specific SME serves as an example for this evaluation.

Inhaltsverzeichnis

1 Einleitung

Datenintegration ist heute in vielen Unternehmen ein sehr aktuelles und meist nicht triviales Thema und das sowohl in großen als auch in kleinen und mittelständischen Unternehmen. Nahezu alle Firmen und Organisationen besitzen große Mengen an Daten, die sie beispielsweise durch Neustrukturierungen, Firmenübernahmen oder aufgrund der heterogenen Bereitstellung von Daten durch Dritte in unterschiedlichsten Formen und Anwendungen speichern. Dies führt oft zwangsläufig zu isolierten, redundanten und inkonsistenten Datenbeständen, was die Integration und Verwaltung erschwert. Um die Einbettung, Administration und Visualisierung der Daten zu erleichtern, stellen viele Software Hersteller Tools zur Verfügung, die entweder ausgewählte kleinere Aufgaben erledigen oder sich gar um alle Business Intelligence Gebiete kümmern [Filbry u.a. 2013 S.13.]. Dabei wird die Anwendungslandschaft hauptsächlich in zwei Bereiche aufgeteilt. Einerseits gibt es die bekannten und umfangreichen, aber auch meist kostspieligen Produkte der großen Firmen wie Oracle, SAP, IBM, etc. Auf der anderen Seite gewinnen Open Source Lösungen von Unternehmen wie Jedox, Pentaho, Talend usw. seit 2008/2009 immer mehr an Relevanz, da diese kostengünstiger sind und meistens einen ausreichend großen Funktionsumfang bieten, was sie vor allem für KMU interessant macht [Filbry u.a. 2013 S.13].

1.1 Problemstellung

Aufgrund der Vielzahl an Tools und dem hohen Kosten- und Zeitaufwand der Datenintegration stellt sich vor allem bei KMU die Frage, welche Anwendungen am besten geeignet sind, um die individuellen Ansprüche des jeweiligen Unternehmens zu erfüllen. Einerseits sollen die Daten schnell, einfach und konsistent eingebettet und verwaltet werden, andererseits soll das verfügbare Budget nicht überschritten und die Datenintegration sowie -administration bestenfalls auch von Laien

problemlos und intuitiv durchgeführt werden können. Insbesondere KMU haben nicht die finanziellen Ressourcen, um teure externe IT- bzw. Datenintegrationsexperten zu engagieren geschweige denn ausreichend eigenes Personal.
Ein weiteres Problem stellt die Heterogenität von zur Verfügung gestellten Daten dar. Wenn ein Unternehmen beispielsweise mehrere verschiedene Lieferanten hat, werden jene ihre Produktdetails in der Regel in unterschiedlichsten Dateiformaten und Schemas bereitstellen. Auch hier muss Software für Datenintegration flexibel sein, um Quelldaten ins korrekte Zielformat zu bringen und somit am Ende einen einheitlichen und konsistenten Datenbestand zu schaffen.

1.2 Ziel

Ziel dieser Bachelorarbeit ist es, einige ausgewählte Tools zur Datenintegration zu evaluieren, die für den Betrieb in KMU geeignet sind. Aufgrund begrenzter finanzieller Mittel in KMU sind daher sehr umfangreiche und vor allem kostspielige Tools von SAP, IBM und Co. ungeeignet. In dieser Arbeit sind folglich nur komplett oder teilweise kostenlose Open Source Lösungen relevant. Letztere reichen auch vom Funktionsumfang her vollkommen aus, da hier lediglich die Integration und Aktualisierung von Datenbeständen behandelt wird und nicht der komplette Business Intelligence Bereich samt Reporting, Analysen usw.

Die Evaluation geschieht exemplarisch am KMU „Ihr Farbraum Metzler & Block GmbH“. Das besagte Unternehmen wurde ca. ein Jahr vor Durchführung dieser Evaluation von zwei neuen Geschäftsführern übernommen, die nun den bestehenden veralteten Online Shop durch einen neuen, größeren und zeitgemäßeren Shop ersetzen wollen, um unter anderem auch den Umsatz zu steigern.
Dazu wird das eCommerce System Shopware eingesetzt, das neben der manuellen Erstellung von Artikeln und Datenbeständen auch den Import aus CSV- und XML-Dateien erlaubt. Im Unternehmen werden hierfür alle benötigten Artikel in der Regel händisch im CSV-Format angelegt und anschließend in Shopware eingepflegt. Da aber manche Lieferanten ihre Artikeldaten auch zur automatisierten Datenverarbeitung beispielsweise als XLSX-Datei zur Verfügung stellen, soll herausgefunden werden, ob sich diese Daten auch maschinell mithilfe von geeigneten Tools importieren und aktualisieren lassen, was sowohl Zeit und somit Kosten spart, als auch die Fehleranfälligkeit durch manuelle Eingaben reduziert. Des Weiteren soll geprüft werden, ob die evaluierten Anwendungen in Zukunft auch für nicht technisch versierte Mitarbeiter intuitiv und zielführend genutzt werden

können, weil die zusätzliche Einstellung von geschultem IT-Personal zu kostenaufwändig wäre.

1.3 Aufbau der Arbeit

Diese Arbeit ist in sieben hier kurz dargestellte Kapitel eingeteilt.

Kapitel 1 gibt eine kurze Einleitung in die Thematik.

Das zweite Kapitel widmet sich dann den Grundlagen, um ein Verständnis für wichtige Begriffe zu schaffen und relevante Dateiquellformate, die ausgewählten Tools sowie das Zielsystem Shopware vorzustellen.

In Kapitel 3 werden die Anforderungen vorgestellt, die die Quelldaten und die beiden Tools erfüllen sollen. Diese Anforderungen bilden auch die Grundlage der Evaluation bzw. der Vergleiche zwischen den Tools.

Kapitel 4 beschreibt die Konzeption des Vergleichs und des ETL Prozesses im Fallbeispiel, sowie als Einschub den Prozess der Datenerhebung.

Das fünfte Kapitel behandelt dann wiederum die Umsetzung des exemplarischen ETL Prozesses in Pentaho und Talend.

In Kapitel 6 folgt die Evaluation, hier wird zunächst die Eignung der Quelldateien bewertet. Anschließend wird ein Urteil über die Tools an sich gefällt und zuletzt ein Fazit gezogen.

Das siebte und zugleich letzte Kapitel fasst schließlich den Inhalt der Arbeit noch einmal zusammen und vermittelt einen kurzen Ausblick.

2 Grundlagen

Dieses Kapitel erörtert grundlegende Begriffe, die für den weiteren Verlauf dieser Ausarbeitung wichtig und unerlässlich sind.
Zunächst gibt Kapitel 2.1 einen kurzen Überblick über Business Intelligence. Kapitel 2.2 erklärt dann relevante Datenquellen. Im Anschluss werden in Kapitel 2.3 die ausgewählten Tools zur Datenintegration vorgestellt ehe sich Kapitel 2.4 mit dem Zielsystem Shopware befasst.

2.1 Business Intelligence

Business Intelligence vereint und ersetzt frühere Begrifflichkeiten wie „Management-Informationssysteme“, „Management-Support-Systeme“ und „IT-basierte Managementunterstützung“. Für den Begriff Business Intelligence, oft auch als „BI“ abgekürzt, gibt es eine Reihe verschiedener Definitionen, da eine Vielzahl unterschiedlicher Technologien unter diesem Begriff zusammengefasst werden.
Einige Definitionen beleuchten dabei eher den technischen Aspekt, wie durch Business Intelligence Anwendungen Daten in Informationen und anschließend Wissen umgewandelt werden, während andere sich mehr auf den wirtschaftlichen Hintergrund beziehen. Das wirtschaftliche Ziel der BI sei die Bereitstellung der richtigen Informationen an die richtigen Leute und zur richtigen Zeit, sodass bessere Entscheidungen getroffen und Wettbewerbsvorteile erlangt werden können [Müller u.a. 2015 S.2, Cebotarean 2011, BSH o.D., Vogel 2016].
Zusammenfassend lässt sich Business Intelligence folglich als eine Sammlung von IT-Komponenten beschreiben, die Daten bzw. Informationen aufbereiten und zur Verfügung stellen, damit aus bestehendem Wissen neue Erkenntnisse gezogen und das Management eines Unternehmens somit bei seinen Entscheidungen unterstützt wird.

In den letzten beiden Jahrzehnten ist BI sowohl im akademischen als auch unternehmerischen Bereich immer wichtiger geworden.
Die Gartner Group fand bspw. 2007 bei einer Umfrage unter 1400 Führungskräften der Informationstechnologie heraus, dass Business Intelligence Projekte die höchste Priorität im Technologiebereich haben [Watson u.a. 2007].
Der IBM Tech Trends Report befragte 2011 über 4000 IT-Experten aus über 93 Ländern und 25 Branchen und machte damit Business Analytik als eines der vier bedeutendsten Technologie Trends der 2010er aus [Chen u.a. 2012].
Außerdem wuchs der Analytik und Business Intelligence Markt im Jahr 2018 um 11,7% auf 21,6 Milliarden US-Dollar. BI Plattformen waren dabei sogar mit 23,3% das am stärksten wachsende Segment [Gupta u.a. 2019].
Business Intelligence und Analytics haben also heute eine große Bedeutung für Unternehmen, weshalb ersteres hier auch vorgestellt wird.
[Abbildung 2.1] zeigt schematisch eine Business Intelligence Architektur. Die erste Schicht der BI Architektur ist die ETL-Schicht, in der Daten aus bereits vorhandenen Systemen bzw. Dateien extrahiert, transformiert und in die zweite Ebene, die Datenspeicherungsschicht geladen wird. Letztere stellt die bspw. in einem Data Warehouse gespeicherten Daten für die dritte Schicht zur Verfügung. In dieser Ebene befinden sich die Anwenderwerkzeuge, die dem Anwender unter anderem durch Reports und Analysen Daten anschaulich bereitstellen und ihn somit z.B. bei Entscheidungen unterstützen [Barc 2013].

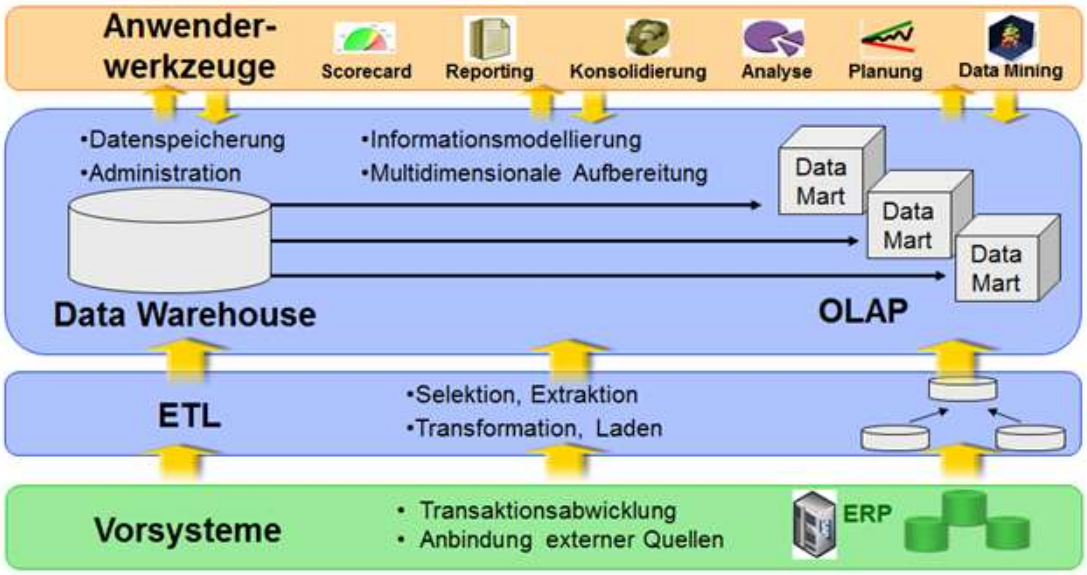


Abbildung 2.1 Schematische Architektur einer Business Intelligence-Lösung [Barc 2013]

Da der Fokus dieser Arbeit nur auf dem Thema Datenintegration liegt, werden andere Bereiche der BI, wie bspw. Big Data Stores, OLAP, Analysen, Reporting und Data Mining sowie die zweite und dritte Schicht der BI Architektur nicht behandelt und somit auch nicht weiter erläutert. Im Folgenden wird dafür aber der für die Datenintegration sehr wichtige Begriff „ETL“ näher erklärt.

2.1.1 Extract Transform Load

Der ETL-Prozess beschreibt mehrere Teilschritte, die bei der klassischen Integration von Daten aus verschiedenen Quellen in ein Zielsystem durchlaufen werden [LUBER 2018].
Die Anfangsbuchstaben dieser Schritte, nämlich „**E**xtract“ (Extrahieren), „**T**ransform“ (Transformieren)“ und „**L**oad“ (Laden) bestimmen daher auch die Namensgebung des **ETL**-Prozesses.
Abbildung 2.2 veranschaulicht den ETL-Prozess.

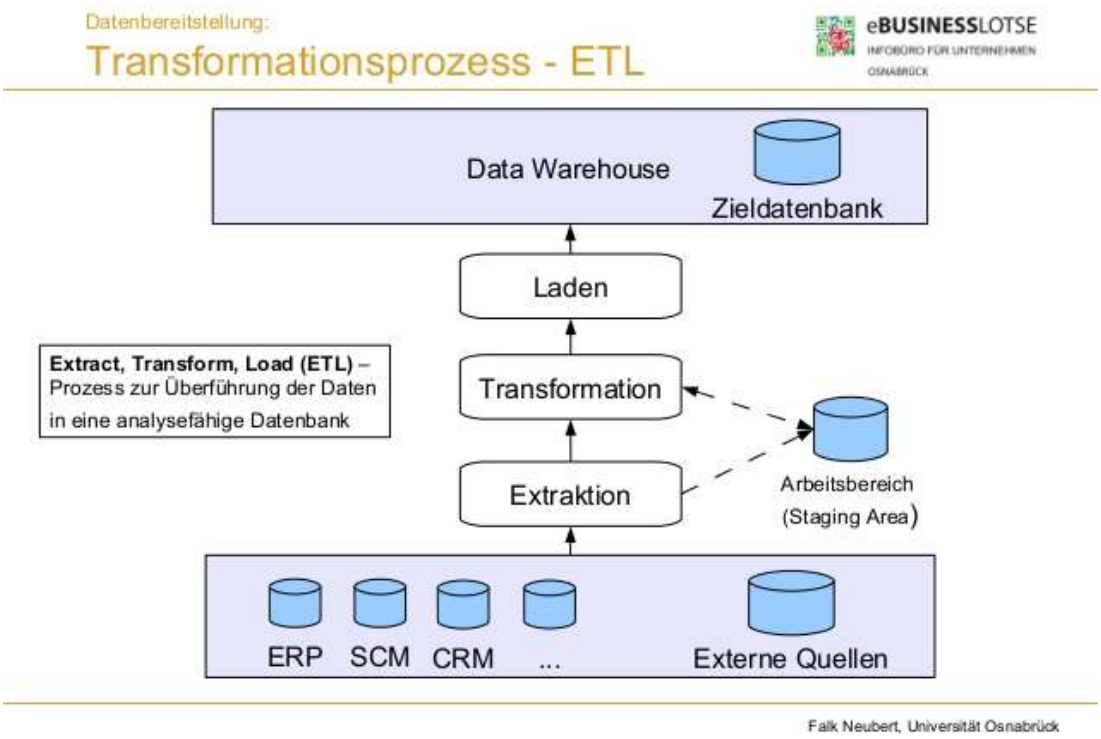


Abbildung 2.2 Transformationsprozess ETL [Neubert 2013]

Zunächst werden im ETL-Ablauf die benötigten Daten aus den vorliegenden Quellen extrahiert (Extract). Dies kann unter Umständen eine große Herausforderung sein,

da die Daten meist in verschiedensten Formaten und Quellsystemen bzw. -dateien vorliegen. So befinden sich manche Daten unter anderem strukturiert in Datenbanken oder Tabellen, stehen semistrukturiert in bspw. JSON-, XML- oder CSV-Dateien zur Verfügung oder sind gar völlig unstrukturiert und müssen zunächst aus Fließtexten wie Mails, Briefen oder Erläuterungen extrahiert werden [Filbry u.a. 2013, Seite 39].
Wenn alle erforderlichen Daten extrahiert wurden, müssen sie im zweiten Schritt transformiert (Transform), also aggregiert und vereinheitlicht werden. Dieser Schritt ist der wichtigste des ETL-Prozesses, aber zugleich auch mit dem höchsten Arbeits- und Zeitaufwand verbunden, da die Daten sowohl syntaktisch (Zeichensätze, Datenformate und -Typen etc.) als auch strukturell (Schlüsselwerte, Constraints, Modellierungen usw.) und gegebenenfalls sogar semantisch (Namensgebungen, Maßeinheiten etc.) homogenisiert werden müssen [Filbry u.a. 2013, Seite 39f].
Nach der erfolgreichen Extraktion und Transformation der Daten, können sie im letzten Schritt ins Zielsystem geladen (Load) werden. Dies kann durch Standardfunktionen der Datenbank geschehen und durch ETL Tools, die den benötigten Sourcecode erzeugen, erleichtert werden. Auch der Load Prozess kann langwierig sein, wenn es viele und oder komplexe Daten zu laden gilt, vor allem das erste bzw. initiale Laden. Spätere Load Prozesse sind meistens weniger aufwändig, da hier normalerweise nicht alle, sondern nur noch veränderte Daten eingespielt werden.

Neben Extract Transform Load (ETL) gibt es auch den Extract Load Transform (ELT) Ansatz. Wie der Name und die veränderte Reihenfolge schon vermuten lässt, werden hierbei die Daten nach dem Extrahieren zuerst ins Zielsystem geladen und dort zuletzt transformiert.
Dies ist z.B. im Big Data Kontext von Vorteil, weil so viele Daten gesammelt und gespeichert werden können und diese erst später verarbeitet werden müssen, um aus ihnen Kenntnisse zu gewinnen. Ein Nachteil ist hier beispielsweise der unzureichende Datenschutz, der durch die Datensammelwut entsteht. Außerdem muss das Zielsystem sehr leistungsfähig sein und darüber hinaus entsprechende Funktionen zur Transformation bereitstellen. Auch die zugehörigen Algorithmen sind in der Regel schwieriger zu programmieren als bei ETL-Anwendungen.
Da es sich im Fallbeispiel nicht um ein Big Data Szenario handelt und das Zielsystem Shopware auch nicht für die Transformation geeignet ist, wird die Beispielanwendung daher ein ETL- und kein ELT-Prozess. Aus diesem Grund wird der ELT-Ansatz hier auch nicht ausführlicher beschrieben.

2.2 Semistrukturierte Daten

Wie bereits in Kapitel 2.1.1 erläutert, gibt es strukturierte, unstrukturierte und semistrukturierte Daten.
Im Rahmen dieser Arbeit stehen nur letztere zur Verfügung, weshalb hier nicht weiter auf strukturierte und unstrukturierte Daten eingegangen wird.
Aus diesem Grund werden in den folgenden drei Unterkapiteln CSV-, XML- und JSON-Dateien vorgestellt.

2.2.1 CSV-Dateien

CSV ist die Abkürzung für „Comma Separated Values“. Dieses Dateiformat, welches konsequenterweise die Dateiendung „.csv“ besitzt, ist ein recht simples Textformat zum Austausch von Daten zwischen (teilweise verschiedenen) Anwendungen und Technologien. Beispiele hierfür sind Datenbanken wie Oracle und MySQL und Programme zur Tabellenkalkulation wie OpenOffice, Numbers oder Microsoft Excel [Debitoor].
Es existiert keine formale Spezifikation des CSV-Formats, beispielsweise werden je nach Implementierung Kommata oder Semikolons als Trennzeichen von Feldern benutzt.
Aus diesem Grund werden im RFC 4180 [Shafranovich 2005] die Eigenschaften beschrieben, die in den meisten Implementierungen vorkommen. Einige der Eigenschaften werden im Folgenden vorgestellt und in Abbildung 2.3 anhand von Beispieldaten veranschaulicht.
Im CSV-Format werden einzelne Datensätze durch einen Zeilenumbruch voneinander getrennt, wobei der letzte Datensatz nicht zwangsläufig einen Zeilenumbruch aufweisen muss. Außerdem kann die erste Zeile einer CSV-Datei zur Benennung von Feldern optional auch eine Header Zeile haben, diese sollte die gleiche Anzahl an Feldern wie die Datensätze haben.
Ein Datensatz besteht wie schon angedeutet aus mindestens einem Feld, mehrere Felder werden durch Kommata separiert. Alle Datensätze sollten gleich viele Felder besitzen [Shafranovich 2005].

```
1  Name,Vorname,Alter,Ort,Strasse,Hausnummer,Postleitzahl
2  Müller,Moritz,23,Hamburg,Hamburger Strasse,3,12345
3  Meier,Tina,21,München,Hauptstrasse,67,67890
4  Schulz,Maria,33,Berlin,Berliner Strasse,45,11111
5  Mustermann,Max,40,Kiel,Dorfstrasse,57,23456
6
7
```

Abbildung 2.3 CSV-Datei mit Beispieldaten

2.2.2 XML-Dateien

XML steht für „Extensible Markup Language“, also eine erweiterbare Sprache für Markierungen. Wie auch bei CSV-Dateien ist die Abkürzung der Extensible Markup Language bezeichnend für die Dateiendung des Dateiformats, nämlich „.xml“.
Eine sehr wichtige Eigenschaft von XML-Dateien ist die Plattformunabhängigkeit, XML funktioniert mit verschiedensten Betriebssystemen wie beispielsweise Windows, Linux und MAC OS sowie auf unterschiedlichen Geräten und Hardwarekonstellationen. Ein System kann also seine Daten in eine XML-Datei exportieren und diese kann wiederum auf beliebig vielen anderen Systemen gelesen und verarbeitet werden, auch wenn jene eine andere Architektur haben.
Die Extensible Markup Language ist ein Framework zur Strukturierung von Daten. Richtlinien hierfür werden z.B. im RFC 3470 [Hollenbeck u.a. 2003] festgelegt. So besitzt XML eine konkrete Syntax, die Zeichen wie „<“, „=“ und „&“ als Trennzeichen nutzt und vom World Wide Web Consortium (W3C) genau beschrieben wird [Bray u.a. 2008]. [Bray u.a. 2008] erläutert beispielsweise auch, wann ein Textdokument ein wohlgeformtes XML-Dokument ist, denn nur genau dann ist es auch XML. Nicht wohlgeformte Zeichenketten und Markierungen sind kein XML [Hollenbeck u.a. 2003].
Abbildung 2.4 demonstriert ein XML-Dokument, dass die gleichen Daten wie die CSV-Datei in Abbildung 2.3 enthält. In der ersten Zeile befindet sich die „XML declaration“, die die XML-Version, nämlich 1.0 und Zeichencodierung, hier UTF-8, enthält.
„Markups“ sind hier beispielsweise „<row>“ und „</row>“ sowie „<Name>“ und „</Name>“, die Anfang und Ende eines Datensatzes respektive Feldes kennzeichnen.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <row>
    <Name>Müller</Name>
    <Vorname>Moritz</Vorname>
    <Alter>23</Alter>
    <Ort>Hamburg</Ort>
    <Strasse>Hamburger Strasse</Strasse>
    <Hausnummer>3</Hausnummer>
    <Postleitzahl>12345</Postleitzahl>
  </row>
  <row>
    <Name>Meier</Name>
    <Vorname>Tina</Vorname>
    <Alter>21</Alter>
    <Ort>München</Ort>
    <Strasse>Hauptstrasse</Strasse>
    <Hausnummer>67</Hausnummer>
    <Postleitzahl>67890</Postleitzahl>
  </row>
  <row>
    <Name>Schulz</Name>
    <Vorname>Maria</Vorname>
    <Alter>33</Alter>
    <Ort>Berlin</Ort>
    <Strasse>Berliner Strasse</Strasse>
    <Hausnummer>45</Hausnummer>
    <Postleitzahl>11111</Postleitzahl>
  </row>
  <row>
    <Name>Mustermann</Name>
    <Vorname>Max</Vorname>
    <Alter>40</Alter>
    <Ort>Kiel</Ort>
    <Strasse>Dorfstrasse</Strasse>
    <Hausnummer>57</Hausnummer>
    <Postleitzahl>23456</Postleitzahl>
  </row>
</root>
```

Abbildung 2.4 XML-Datei mit den gleichen Datensätzen wie in Abb. 2.3

2.2.3 JSON-Dateien

JSON ist die Abkürzung für „JavaScript Object Notation“. Wie der Name schon vermuten lässt, wurde dieses leichtgewichtige Datenformat von der Skriptsprache JavaScript abgeleitet und kann daher mit dieser in der Regel auch interpretiert werden. Darüber hinaus ist JSON einfach zu lesen und zu schreiben, sowohl für Menschen als auch für Computer. Mit JSON können primitive Daten wie Strings, Numbers, Booleans und „null“ sowie strukturierte Typen wie Objekte und Arrays gespeichert und transportiert werden. Dabei ist JSON plattform- und programmiersprachenunabhängig und wird daher oft zum Datenaustausch zwischen Anwendungen oder zwischen Server und Website genutzt [Bray 2017, w3schools o.D., Augsten 2018].

Daten in JSON werden in „name-value pairs“ gespeichert, beide Teile werden durch einen Doppelpunkt separiert. Ein Komma trennt wiederum die einzelnen „name-

value pairs“ voneinander. Daten in geschweiften Klammern bilden in JSON ein Objekt, während Array Werte in eckigen Klammern gespeichert werden.
Für eine ausführliche Beschreibung der JSON-Syntax und des JSON-Formats an sich sei hier nochmal auf RFC 8259 [Bray 2018] verwiesen.


```
[
  {
    "Name": "Müller",
    "Vorname": "Moritz",
    "Alter": 23,
    "Ort": "Hamburg",
    "Strasse": "Hamburger Strasse",
    "Hausnummer": 3,
    "Postleitzahl": 12345
  },
  {
    "Name": "Meier",
    "Vorname": "Tina",
    "Alter": 21,
    "Ort": "München",
    "Strasse": "Hauptstrasse",
    "Hausnummer": 67,
    "Postleitzahl": 67890
  },
  {
    "Name": "Schulz",
    "Vorname": "Maria",
    "Alter": 33,
    "Ort": "Berlin",
    "Strasse": "Berliner Strasse",
    "Hausnummer": 45,
    "Postleitzahl": 11111
  },
  {
    "Name": "Mustermann",
    "Vorname": "Max",
    "Alter": 40,
    "Ort": "Kiel",
    "Strasse": "Dorfstrasse",
    "Hausnummer": 57,
    "Postleitzahl": 23456
  }
]
```


Abbildung 2.5 JSON-Datei mit den gleichen Datensätzen wie in Abb. 2.3 und 2.4

2.3 Tools zur Datenintegration

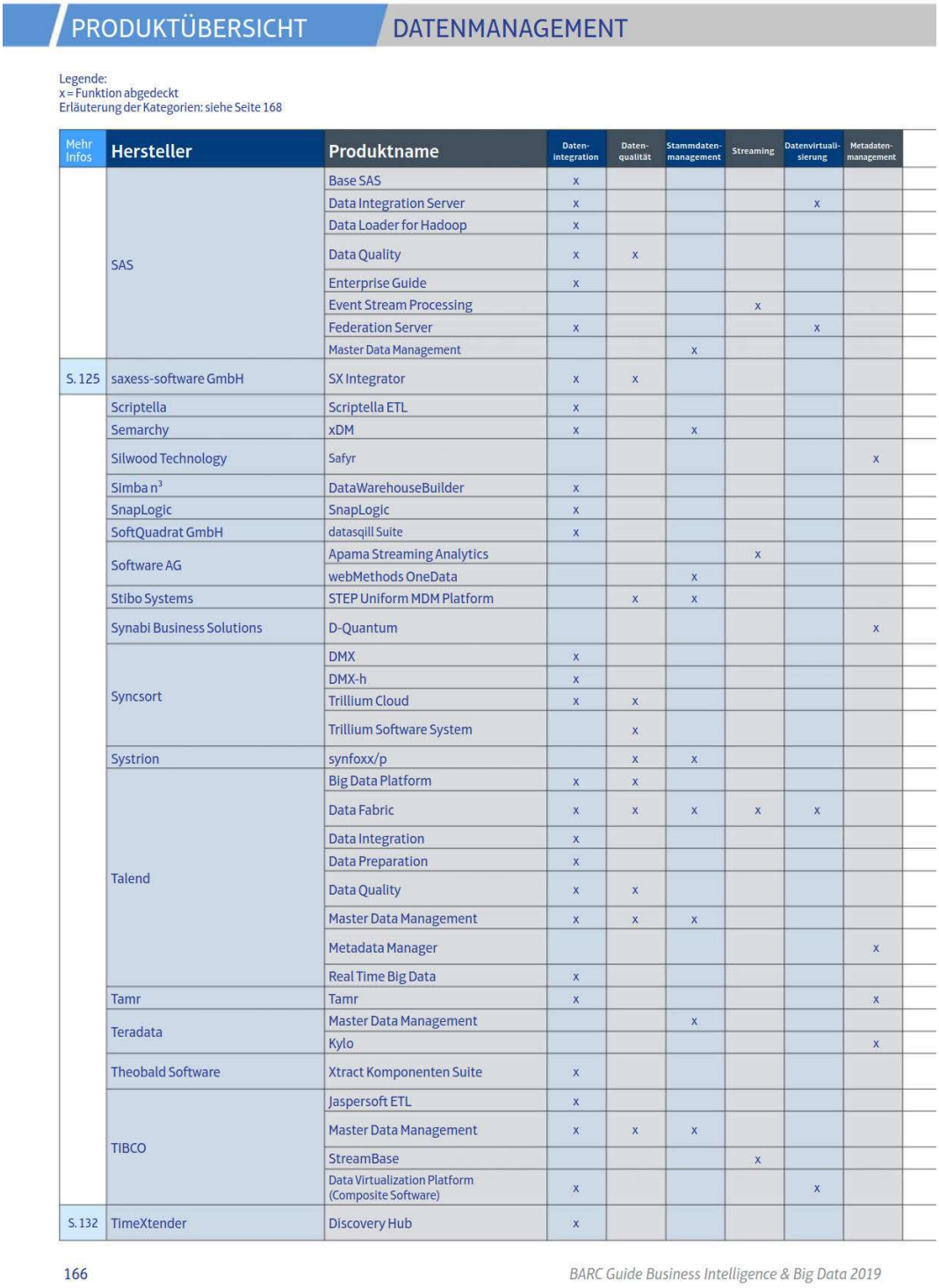
Auf dem Markt findet man eine Vielzahl an Tools zur Datenintegration, die beispielsweise als Fragment einer BI Suite, eigenständiges Tool oder Erweiterung einer Datenbank vorkommen. Eine Auswahl dieser Tools ist in den folgenden Tabellen 2.1 und 2.2 ersichtlich.
Die in dieser Arbeit zu evaluierenden Werkzeuge sollen folgende Kriterien erfüllen:

(I) Keine Kosten für Download, Installation und Nutzung bei moderaten Datenmengen und Basisfunktionen
(II) Plattformunabhängigkeit: Betrieb auf den Betriebssystem Windows und MacOS soll möglich sein
(III) Mindestens ein Tool soll zu den großen führenden Anbietern für Datenintegration gehören, das zweite kann ein Nischenprodukt sein

Legende:
x = Funktion abgedeckt
Erläuterung der Kategorien: siehe Seite 168

Mehr Infos	Hersteller	Produktname	Daten-integration	Daten-qualität	Stammdaten-management	Streaming	Datenvirtuali-sierung	Metadaten-management
	Oracle	Big Data Preparation Cloud Service	x	x				
		Data Integrator	x					
		Data Profiling		x				
		Data Service Integrator					x	
		Edge Analytics				x		
		Enterprise Data Quality		x				
		Enterprise Metadata Management						x
		GoldenGate	x					
		Hyperion Data Relationship Management			x			
		Master Data Management	x	x	x			
		Stream Analytics				x		
		Warehouse Builder	x					
	Orchestra Networks	EBX5			x			
	Paxata	Paxata	x					
	Pentaho (Hitachi Vantara)	Pentaho Data Integration (Community und Enterprise Edition)	x					
	Pitney Bowes	Sagent Data Flow	x					
		Spectrum		x				
	PRODATA	AdressExpert		x				
	Profisee	Maestro		x	x			
	Prospero	PRO-NC Namechecking		x				
	PST	PST-Data Warehouse	x		x			
	Quadient	DataCleaner (Cloud)		x				
		DataEntry		x				
		Datahub		x	x			
	Record Evolution	Repods	x					
	Redhat	JBoss Middleware	x				x	
	Reply	Arlanis Integrations-Platform	x					
	Rosslyn Analytics	Rapid Platform	x	x				
	SAP	BusinessObjects Information Steward						x
		Data Integrator	x					
		Data Quality Management		x				
		Data Services	x	x				
		HANA Information Management Option	x	x			x	
		Information Steward		x	x			
		Master Data Governance (SAP MDG)			x			
		Netweaver Master Data Management (SAP Netweaver MDM)			x			
		Sybase Event Stream Processor				x		
		Sybase PowerDesigner						x

Tabelle 2.1 Auszug (1) der Datenmanagement Produkte aus [Barc 2019]



PRODUKTÜBERSICHT DATENMANAGEMENT

Legende:
x = Funktion abgedeckt
Erläuterung der Kategorien: siehe Seite 168

Mehr Infos	Hersteller	Produktname	Daten-integration	Daten-qualität	Stammdaten-management	Streaming	Datenvirtuali-sierung	Metadaten-management
	SAS	Base SAS	x					
		Data Integration Server	x				x	
		Data Loader for Hadoop	x					
		Data Quality	x	x				
		Enterprise Guide	x					
		Event Stream Processing				x		
		Federation Server	x				x	
		Master Data Management			x			
S. 125	saxess-software GmbH	SX Integrator	x	x				
	Scriptella	Scriptella ETL	x					
	Semarchy	xDM	x		x			
	Silwood Technology	Safyr						x
	Simba n³	DataWarehouseBuilder	x					
	SnapLogic	SnapLogic	x					
	SoftQuadrat GmbH	datasqill Suite	x					
	Software AG	Apama Streaming Analytics				x		
		webMethods OneData			x			
	Stibo Systems	STEP Uniform MDM Platform		x	x			
	Synabi Business Solutions	D-Quantum						x
	Syncsort	DMX	x					
		DMX-h	x					
		Trillium Cloud	x	x				
		Trillium Software System		x				
	Systrion	synfoxx/p		x	x			
	Talend	Big Data Platform	x	x				
		Data Fabric	x	x	x	x	x	
		Data Integration	x					
		Data Preparation	x					
		Data Quality	x	x				
		Master Data Management	x	x	x			
		Metadata Manager						x
		Real Time Big Data	x					
	Tamr	Tamr	x					x
	Teradata	Master Data Management			x			
		Kylo						x
	Theobald Software	Xtract Komponenten Suite	x					
	TIBCO	Jaspersoft ETL	x					
		Master Data Management	x	x	x			
		StreamBase				x		
		Data Virtualization Platform (Composite Software)	x				x	
S. 132	TimeXtender	Discovery Hub	x					

166 BARC Guide Business Intelligence & Big Data 2019

Tabelle 2.2 Auszug (2) der Datenmanagement Produkte aus [Barc 2019]

Die Tabellen 2.1 und 2.2 zeigen einen Auszug von Datenmanagement Produkten und demonstrieren u.a., welche Tools die Datenintegration beherrschen. Die gesamte Übersicht ist zu groß und umfangreich, um sie hier komplett darzustellen und näher zu betrachten. Im BARC Guide Business Intelligence & Big Data 2019 [BARC 2019] ist die komplette Tabelle zu finden.
In den Tabellen 2.1 und 2.2 finden sich u.a. auch die Werkzeuge Talend Data Integration und Pentaho Data Integration. Beide sind Open Source Lösungen und bieten daher im Gegensatz zu manchen anderen Produkten wie SAP Data Services auch eine kostenlose Basisversion an, die für KMU gegebenenfalls ausreichend ist und das verfügbare Budget nicht durch den Kauf von weiterer Software belastet. Anforderung (I) wird also von beiden Tools erfüllt.
Des Weiteren können Pentaho Data Integration und Talend Data Integration sowohl auf Mac OS als auch auf Windows System betrieben werden, genügen also auch Anforderung (II).

Abbildung 2.6 Magic Quadrant for Data Integration Tools [Gartner 2018]

Abbildung 2.6 stellt das Magic Quadrant for Data Integration Tools von [Gartner 2018] dar. In diesem werden ausgewählte Datenintegrationswerkzeuge in die vier Kategorien bzw. Quadranten Nischenanbieter, Visionäre, Herausforderer und Marktführer eingeordnet. Talend wird hierbei als einer der Marktführer eingestuft, Pentaho bzw. der Mutterkonzern Hitachi Vantara als ein Nischenanbieter. Daher erfüllen beide Tools auch Anforderung (III).
Da Talend Data Integration und Pentaho Data Integration allen Anforderungen (I), (II) und (III) genügen, werden beide Anwendungen als Gegenüberstellung im Rahmen dieser Arbeit evaluiert und im Folgenden näher vorgestellt.

2.3.1 Pentaho

Pentaho wurde von der 2004 gegründeten Pentaho Corporation entwickelt und 2015 vom Unternehmen Hitachi Data Systems übernommen.
Pentaho ist eine Ansammlung unterschiedlicher Business Intelligence Tools. Diese beinhaltet ETL, Reporting, Data-Mining, Big Data, Datenintegration, -visualisierung und -analyse.
Darüber hinaus wird Pentaho sowohl in einer kostenpflichtigen Enterprise Version mit mehr Funktionen und professionellem Support als auch in einer kostenlosen Open Source Community Edition angeboten, die Basisfunktionen enthält und nur durch Support von der Pentaho Community unterstützt wird. Im Open Source Bereich gehört Pentaho seit über 10 Jahren zu den wichtigsten und bekanntesten BI Lösungen [LUBER 2017].
Da in dieser Arbeit kostengünstige Tools zur Datenintegration evaluiert werden sollen, beschränkt sich diese Bewertung auf die Datenintegrations-Komponente der kostenlosen Pentaho Community Edition. Die kostenpflichtige Enterprise Edition sowie die anderen Bereiche der Pentaho BI Software werden hier nicht näher betrachtet.

2.3.2 Talend

Talend ist eine Anwendung zur Datenintegration, die vom gleichnamigen kalifornischen Unternehmen entwickelt und 2006 vorgestellt wurde.
Im Gegensatz zu Pentaho beschränkt sich Talend auf Integration und Management von Daten, eine umfangreiche BI Suite wird nicht angeboten.
Ebenso wie Pentaho existiert bei Talend neben der kostenlosen Open Source Lösung Open Studio for Data Integration mit Talend Enterprise Data Integration eine kostenpflichtige Enterprise Lösung [Müller u.a. 2015 S.25, Filbry u.a. 2013, Goram o.D., Sherman 2016, Talend 2019 I].

Für KMU ist aufgrund des Kostenfaktors die frei verfügbare Open Source Lösung Open Studio am relevantesten, weshalb im weiteren Verlauf der Arbeit nur diese Version evaluiert wird.

2.4 Shopware

Das modulare Online-Shopsystem Shopware wurde erstmals 2003 von den Hamann Brüdern in Deutschland entwickelt. 2008 folgte dann zusammen mit Stefan Heyne die Gründung der Shopware AG. Große Popularität – Shopware ist eines der bekanntesten Online-Shopsysteme im deutschsprachigen Raum - erlangte Shopware im Jahr 2010, als erstmals eine Open Source basierte Community Edition veröffentlicht wurde [Shopware o.D., Mittwald o.D.].

Auch Shopware wird sowohl in einer kostenlosen Open Source Version als auch in kommerziellen Enterprise bzw. Professional Software Suites angeboten.

Um eine große Anzahl von Artikeln nicht manuell eingeben zu müssen, stellt Shopware einen komfortableren Import von Daten aus externen Quellen wie XML- oder CSV-Dateien zur Verfügung [Shopware 2019].

In dieser Arbeit werden geeignete Software Lösungen evaluiert, die aus beispielhaften JSON-, XML- und CSV-Dateien Daten extrahieren und in ein für den Shopware Import geeignetes Zielformat transformieren.

3 Anforderungen

Dieses Kapitel widmet sich den Anforderungen. Dazu wird zunächst in 3.1 analysiert, welche Anforderungen an die Daten gestellt werden. Anschließend werden in 3.2 Anforderungen an die Software, also die beiden Tools erarbeitet. Im späteren Verlauf dieser Arbeit wird dann überprüft, welche der Anforderungen die Daten bzw. die Tools erfüllen, um somit eine Bewertung vornehmen zu können.

3.1 Anforderungen an Daten

Im Folgenden werden die Anforderungen an Daten formuliert, wobei zwischen methodischen und in der Fallstudie relevanten Anforderungen unterschieden wird.

3.1.1 Methodik

Es soll gezeigt werden, dass Pentaho und Talend Transformationen mit Daten aus unterschiedlichen Quellformaten durchführen können. Daher sollen hier exemplarisch Daten aus den vier verschiedenen Dateiformaten XML, CSV, JSON und auch XLSX importiert werden.
Inkonsistenzen in den Daten würden bei der Evaluation beider Tools helfen, um herauszufinden, ob und wenn ja wie sie diese beheben können. Daher sind sie bei mindestens einer der Dateien sogar gewünscht.
Das Zielformat nach den Transformationen soll jeweils das CSV-Format sein.

3.1.2 Fallstudie

Eine der Quelldateien muss Artikel samt Artikelinformationen enthalten, die in den neuen Online Shop von „Ihr Farbraum Metzler & Block“ integriert werden sollen. Dies könnten beispielsweise bei einem Lack der Name und Preis, die Artikelbeschreibung und die Anwendungshinweise sein. Zudem sollen je nach

Artikel Variationen vorliegen, also etwa die verschiedenen Farbtöne und (Gebinde-) Größen.
Die anderen Dateien können simple Beispieldaten wie Wetterdaten enthalten, da der aufwendige Transformationsprozess der Artikeldaten zur Evaluation nur einmal durchgeführt werden muss. Bei gleichen Daten trotz verschiedener Quelldateiformate wäre dieser nach dem erfolgreichen Import nämlich analog.

3.2 Anforderungen an Software

Kapitel 3.2 beschreibt die Anforderungen, die an die Software bzw. die Tools gestellt werden. Auch hier wird analog zu den Anforderungen an Daten zwischen Methodik und Fallstudie differenziert.

3.2.1 Methodik

An die beiden Tools Pentaho und Talend werden eine Reihe funktionaler und nichtfunktionaler Anforderungen gestellt. Anhand dieser Anforderungen soll evaluiert werden, wie gut beide Anwendungen für die Datenintegration geeignet sind und welche ggf. besser zum KMU „Ihr Farbraum Metzler & Block“ passt.
Im Folgenden werden funktionale und nichtfunktionale Anforderungen an die Software vorgestellt und anhand der MoSCoW Methode [Ahmad u.a. 2017] priorisiert. Hierbei steht **(M)** für Must Have, also zwingend erforderliche Anforderungen, **(S)** für Should Have, also Anforderungen, die das Tool erfüllen sollte und **(C)** für Could Have, d.h. Anforderungen, die die Anwendung bestenfalls haben könnte, sofern **(M)** oder **(S)** Anforderungen dadurch nicht beschränkt werden.
(W), also Would Have Anforderungen existieren hier nicht.

Um die zu evaluierenden Tools auf ihre Fähigkeiten im Bereich Datenintegration zu testen und um ihre Eignung für den ETL Prozess zu prüfen, wurden folgende funktionale Anforderungen abgeleitet:

(FA1) XLSX-Dateien können geladen und die Daten darin verarbeitet werden. Beispielhaft soll dazu eine XLSX Datei mit Artikeldaten importiert werden. **(M)**
(FA2) CSV-Dateien wie in 2.2.1 beschrieben sollen importiert werden können. Exemplarisch soll hier die in Kapitel 4.2 beschriebene CSV-Datei mit Wetterdaten geladen werden. **(M)**
(FA3) Das Einlesen von wohlgeformten XML-Dateien wie in Kapitel 2.2.2 soll möglich sein. Auch hier wird als Beispiel der Import einer in 4.2 vorgestellten XML-Datei durchgeführt. **(M)**

(FA4) Analog zu (FA2) muss der Import von JSON-Dateien (Kapitel 2.2.3) möglich sein, was ebenfalls exemplarisch an einer JSON-Datei geschieht, die in 4.2 erläutert wird. **(M)**

(FA5) Transformation der Daten ins benötigte Zielformat. Im Fallbeispiel sollen dazu die Daten wie Gebindegröße, Farbton etc. aus den Quelldateien so transformiert werden, dass sie am Ende eine vorgegebene Struktur aufweisen, die dann als CSV-Datei ins Shopware System geladen werden kann. **(M)**

(FA6) Laden/Exportieren der transformierten Daten in jeweils eine CSV-Datei wie in 2.2.1 **(M)**

(FA7) Fehlertoleranz/-behebung bei inkonsistenten Datensätzen. **(S)**

Ergänzt werden diese funktionalen durch folgende nichtfunktionale Anforderungen, die aus [Glinz 2006] abgeleitet wurden:

(NFA1) Funktionalität - Korrektheit (Richtigkeit):
Das jeweilige Tool durchläuft den ETL Prozess. Es extrahiert, transformiert und lädt alle Datensätze aus den Quelldateien korrekt und vollständig. **(M)**

(NFA2) Benutzbarkeit - Bedienbarkeit:
Das Tool ist auch von jemandem bedienbar, der sich vorher nicht weiter damit befasst hat, also Import einer neuen/aktualisierten Quelldatei und Ausführung des ETL Prozesses sind für ihn leicht durchführbar. **(S)**

(NFA3) Effizienz - Zeitverhalten:
Start des Programms und Ausführung des ETL Prozesses sollen insgesamt maximal fünf Minuten dauern. Daher soll das Tool zum vollständigen Start (Start des Programms & Start des Projekts) maximal vier Minuten und zur Durchführung der Transformation maximal eine Minute beanspruchen. Kürzere Zeiten wären auch wünschenswert. **(S)**

(NFA4) Benutzbarkeit - Verständlichkeit:
Der finale ETL Prozess sowie das Tool sind auch für Neulinge einfach zu verstehen. **(C)**

(NFA5) Benutzbarkeit – Erlernbarkeit:
Ein Nicht-ITler kann das Tool erlernen, also den vorhandenen ETL Prozess modifizieren. Außerdem kann er selbst ETL Prozesse für andere Import Dateien mithilfe des Tools entwickeln. **(C)**

3.2.2 Fallstudie

Zur Überprüfung der Anforderungen werden exemplarisch einige Daten erhoben, transformiert und exportiert. Der Prozess der Datenerhebung wird in Kapitel 4.2 näher erläutert. Für (NFA2), (NFA4) und (NFA5) wird beispielhaft ein Mitarbeiter des Betriebs „Ihr Farbraum Metzler & Block“ herangezogen.

4 Konzeption

In diesem Kapitel wird zunächst konzipiert, wie beide Tools verglichen werden können, um später eine Bewertung zu ermöglichen (4.1).
Für den Vergleich werden Beispielanwendungen bzw. -prozesse in Talend und Pentaho entworfen und umgesetzt. Dafür wurden zunächst Daten erhoben wie Kapitel 4.2 auch näher beschreibt.
Zuletzt erfolgt in 4.3 die Konzeption des ETL Prozess für die Artikeldaten der Marke Volvox, die wie in 4.2 beschrieben gesammelt wurden und in den neuen Online Shop integriert werden sollen.
Der ETL Prozess für die ebenfalls in 4.2 erhobenen Wetterdaten wird nicht konzeptioniert, da hier hauptsächlich zu zeigen ist, dass beide Tools in der Lage sind, CSV-, XML- sowie JSON-Dateien zu laden. Es erfolgt hier also kein umfangreicher Transformationsprozess.

4.1 Konzeption des Vergleichs

Im Folgenden wird der Vergleich zwischen beiden Tools konzeptioniert.
Auch die Konzeption wird hier nach Methodik und Fallbeispiel unterschieden, da für ein KMU in einem spezifischen Anwendungsfall verständlicherweise andere Prioritäten gesetzt werden und manche Funktionen wichtiger oder weniger wichtig sind als im allgemeinen Fall.

4.1.1 Methodik

Um Pentaho und Talend zu vergleichen, muss evaluiert werden, ob und inwieweit beide Tools die Anforderungen aus 3.2.1 erfüllen.
Bei den funktionalen Anforderungen muss zusätzlich getestet werden, ob sie alle in (FA1) bis (FA4) beschriebenen Dateiformate importieren können. Außerdem muss

in beiden Tools ein Transformationsprozess programmierbar sein, der die Daten ins korrekte Format überführt (FA5) und danach in jeweils eine CSV-Datei exportiert (FA6). Bei anfangs inkonsistenten Datensätzen muss überprüft werden, ob diese am Ende von beiden Tools behoben wurden.
Auch die nichtfunktionalen Anforderungen müssen für den Vergleich herangezogen werden. Die jeweils resultierenden CSV-Dateien werden hierfür auf Gleichheit, Vollständigkeit und Korrektheit überprüft (NFA1).
Zudem muss analysiert werden und verglichen werden, welches Tool geeigneter für Nicht-ITler bzw. Neulinge ist. Dazu wird die Erfüllung der nichtfunktionalen Anforderungen (NFA2), (NFA4) und (NFA5) kontrolliert.
Des Weiteren wird die Performance beider Tools (NFA3) stichprobenartig gegenübergestellt. Dazu werden beide Tools mehrmals gestartet, sowie der jeweilige ETL Prozess ausgeführt. Dabei wird dann jeweils der Zeitaufwand gemessen.
Für all diese Vergleiche werden Beispielanwendungen bzw. -ETL-Prozesse in jeweils beiden Tools programmiert. Einerseits kleine Prozesse zur Validierung von (FA1) bis (FA4) und andererseits ein aufwändigerer Prozess zur Überprüfung der übrigen Anforderungen.

4.1.2 Fallstudie

Für den Vergleich beider Tools anhand der Fallstudie und um herauszufinden, welches von ihnen besser für den Betrieb „Ihr Farbraum Metzler & Block“ geeignet ist, werden insbesondere (NFA2) bis (NFA5) analysiert. Diese werden wie alle anderen nichtfunktionalen und funktionalen Anforderungen zwar schon vorher überprüft, allerdings ist beim Praxisbeispiel insbesondere von Bedeutung, welches Tool besser für Mitarbeiter des Betriebs „Ihr Farbraum Metzler & Block geeignet ist, da diese alle wenig IT Kenntnisse besitzen und in Zukunft ggf. auch selbst die Prozesse ausführen, anpassen oder neue Quelldateien importieren sollen, wenn sich beispielsweise Preise geändert haben.
Zudem soll der Start des Tools sowie die Ausführung des ETL Prozesses möglichst wenig Zeit in Anspruch nehmen.

4.2 Datenerhebung

Um an Daten für die Artikel im neuen Online Shop zu kommen, wurden zunächst mit den betroffenen Lieferanten Telefonate geführt. Dabei wurde u.a. gefragt, ob eine Liste mit allen Artikeln des jeweiligen Herstellers beispielsweise in einer Datenbank oder aber zumindest in lokalen Dateien wie XML-, CSV-, oder XLSX-

Dateien vorliegen und ob der Firma Metzler & Block diese zur Verfügung gestellt werden könnten.
Dies war leider nicht bei vielen Herstellern der Fall, weshalb die Daten meist händisch angelegt werden mussten.
Glücklicherweise kam bei den Telefonaten aber heraus, dass die Artikel der Marke Volvox in einer XLSX-Datei vorliegen und das KMU Metzer & Block diese für den automatisierten Import in den neuen Online Shop nutzen kann und darf, sofern es sich selbst um Extraktion, Transformation und Laden der Daten kümmert.

	Artikel-Nr.	Bezeichnung1	Bezeichnung2	Gewicht	Gewi	Empf. VK, netto	Empf. VK, brutto	EAN
887	3-1722	VOLVOX Clenovo Lehmfarbe weiß 5 l	ohne Konservierungsm	8,000	kg	52,61 €	62,60 €	4029955317229
888	3-1723	VOLVOX Clenovo Lehmfarbe weiß 10 l	ohne Konservierungsm	16,000	kg	87,48 €	104,10 €	4029955317236
889	3-1726	VOLVOX Clenovo Lehmfarbe weiß 0,9 l	ohne Konservierungsm	1,440	kg	14,54 €	16,50 €	4029955317267
890	3-1729	VOLVOX Clenovo Lehmfarbe weiß 0,1 l	ohne Konservierungsm	0,160	kg	3,11 €	3,70 €	4029955317298
891	3-1800	VOLVOX feineErde Lehmedelputz	creme, 500 g, VE 18	500,000	g	3,28 €	3,90 €	4029955318004
892	3-1804	VOLVOX feineErde Lehmedelputz	creme, 25 kg, VE 40	25,500	kg	51,60 €	61,40 €	4029955318042
893	3-1810	VOLVOX feineErde Lehmedelputz	zimt, 500 g, VE 18	550,000	g	3,28 €	3,90 €	4029955318103
894	3-1814	VOLVOX feineErde Lehmedelputz	zimt, 3/000, 25 kg, VE	25,500	kg	51,60 €	61,40 €	4029955318141
895	3-1820	VOLVOX feineErde Lehmedelputz	weiß, 500 g, VE 18	500,000	g	3,28 €	3,90 €	4029955318202
896	3-1824	VOLVOX feineErde Lehmedelputz	weiß, 3/000, 25 kg, Vl	25,500	kg	51,60 €	61,40 €	4029955318240
897	3-1830	VOLVOX feineErde Lehmedelputz	ocker, 500 g, VE 18	500,000	g	3,28 €	3,90 €	4029955318301
898	3-1834	VOLVOX feineErde Lehmedelputz	ocker, 3/400, 25 kg, V	25,500	kg	51,60 €	61,40 €	4029955318349
899	3-1840	VOLVOX feineErde Lehmedelputz	orange, 500 g, VE 18	500,000	g	3,28 €	3,90 €	4029955318400
900	3-1844	VOLVOX feineErde Lehmedelputz	orange, 3/300, 25 kg,	25,500	kg	51,60 €	61,40 €	4029955318448
901	3-1880	VOLVOX feineErde Lehmedelputz	terracotta, 500 g, VE	500,000	g	3,28 €	3,90 €	4029955318806
902	3-1884	VOLVOX feineErde Lehmedelputz	terracotta, 3/100, 25 k	25,500	kg	51,60 €	61,40 €	4029955318844
903	3-1890	VOLVOX feineErde Lehmedelputz	sahara, 500, VE 18	500,000	g	3,28 €	3,90 €	4029955318905
904	3-1894	VOLVOX feineErde Lehmedelputz	sahara, 25 kg, VE 40	25,500	kg	51,60 €	61,40 €	4029955318943
905	3-2011	VOLVOX Espressivo Lehmfarbe	lychée (B), 2,5 l, VE=3	3,500	kg	31,76 €	37,80 €	4029955320113
906	3-2012	VOLVOX Espressivo Lehmfarbe	lychée (B), 5 l, VE=3	7,500	kg	56,22 €	66,90 €	4029955320120
907	3-2013	VOLVOX Espressivo Lehmfarbe	lychée (B), 10 l, VE=6	15,000	kg	94,37 €	112,30 €	4029955320137
908	3-2016	VOLVOX Espressivo Lehmfarbe	lychée (B), 0,9 l, VE=3	1,500	kg	15,71 €	18,70 €	4029955320168
909	3-2019	VOLVOX Espressivo Lehmfarbe	lychée (B), 100 ml, VE	0,170	kg	2,69 €	3,20 €	4029955320199
910	3-2021	VOLVOX Espressivo Lehmfarbe	ice (B), 2,5 l, VE=3	3,500	kg	31,76 €	37,80 €	4029955320212
911	3-2022	VOLVOX Espressivo Lehmfarbe	ice (B), 5 l, VE=3	7,500	kg	56,22 €	66,90 €	4029955320229
912	3-2023	VOLVOX Espressivo Lehmfarbe	ice (B), 10 l, VE=6, Pa	15,000	kg	94,37 €	112,30 €	4029955320236
913	3-2026	VOLVOX Espressivo Lehmfarbe	ice (B), 0,9 l, VE=3	1,500	kg	15,71 €	18,70 €	4029955320267
914	3-2029	VOLVOX Espressivo Lehmfarbe	ice (B), 100 ml, VE=4	0,170	kg	2,69 €	3,20 €	4029955320298
915	3-2031	VOLVOX Espressivo Lehmfarbe	café au lait (B), 2,5 l, '	3,500	kg	31,76 €	37,80 €	4029955320311

Tabelle1

Tabelle 4.1 Auszug aus der Artikel.xlsx von Volvox

Tabelle 4.1 zeigt einen Auszug aus der „Artikel.xlsx“ Datei, die von der „Ecotec Naturfarben GmbH“ zur Verfügung gestellt wurde und neben Produktnamen beispielsweise Farbtöne, Gebindegrößen und EAN Codes zu Produkten der Marke „Volvox“ enthält.
Für den ETL Prozess werden lediglich die Spalten „Bezeichnung1“, die den Produktnamen beinhaltet, „Bezeichnung2“, die unter anderem Gebindegröße und Farbton angibt, sowie „EAN“ benötigt.
Außerdem werden später für den ETL Prozess lediglich alle Zeilen mit dem Artikel „Espressivo Lehmfarbe“ benötigt. Einerseits gibt es den Artikel nämlich in über 800 Variationen, was für die Evaluation ausreichend ist und andererseits möchte das KMU „Ihr Farbraum Metzler & Block“ lediglich diesen Artikel von Volvox im Online Shop verkaufen. Weitere Produkte könnten analog eingepflegt werden, hätten somit aber dementsprechend für diese Bachelorarbeit keinen Mehrwert

Des Weiteren wurden teilweise darüber hinaus benötigte Daten wie vor allem Beschreibungstexte für das Produkt „Espressivo Lehmfarbe“ händisch aus anderen Quellen, insbesondere von der Website des Herstellers bezogen und in eine „Import_Volvox.xlsx“ Datei geschrieben.

	A	B	C	BW	BX	BY
1	ordernumb	mainnun	name	Zusammenfassung	Beschreibung	Anwendung
2	VVX0001	VVX0001	Volvox Espressivo Lehmfarbe	Lösemittelfreier,	Volvox feineErde Lehmfarbe	Für deckende und
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						

Tabelle 4.2 Auszug aus der selbst erstellten Import_Volvox.xlsx

In Tabelle 4.2 ist eine kleine Auswahl der zahlreichen Spalten der „Import_Volvox.xlsx“ Datei zu sehen. Diese einzeilige Datei enthält die allgemeinen Beschreibungen und Informationen für den benötigten Artikel. Daten wie die Produktbeschreibung oder Anwendungshinweise gelten für alle Variationen, d.h. Kombinationen von Gebindegröße und Farbe, und müssen aus diesem Grund später in jede Zeile, also jeden Datensatz der Zieldatei geschrieben werden.
Darüber hinaus besitzt diese .xlsx Datei bereits die gewünschte Zielstruktur, die am Ende des ETL Prozesses vorliegen muss, um in Shopware importiert werden zu können.

Da neben der XLSX- auch jeweils eine JSON-, XML- und CSV-Datei zur Evaluation der Tools importiert werden soll, wurden Wetterdaten für die Stadt Hamburg über die API von OpenWeatherMap [OpenWeather 2019] heruntergeladen. Auf dieser Seite kann man nach kostenloser Registrierung mithilfe eines API Keys die Wetterdaten für viele Städte auf der ganzen Welt abrufen. Standardmäßig erhält man die Daten dann im JSON-Format, mit einem modifizierten API-Call kann man sie aber auch als XML-Datei erhalten.
Leider bietet die OpenWeatherMap keine Wetterdaten im CSV-Format. Allerdings kann als Beispieldatei für den Import auch einfach eine der aus den anderen Transformationen resultierenden Dateien genommen werden, da laut (FA6) das Zielformat immer eine CSV-Datei ist.

```
city;coord_lon;coord_lat;weather_id;weather_main;
Hamburg;10.0;53.55;800;Clear;clear sky;01d;statio

```

Abbildung 4.1 Wetterdaten für Hamburg im CSV-Format [OpenWeather 2019]

```
<current>
<city id="2911298" name="Hamburg">
 <coord lon="10" lat="53.55"/>
 <country>DE</country>
 <timezone>7200</timezone>
 <sun rise="2019-10-18T05:51:38" set="2019-10-18T16:18:29"/>
</city>
<temperature value="290.07" min="288.71" max="291.48" unit="kelvin"/>
<humidity value="67" unit="%"/>
<pressure value="1002" unit="hPa"/>
<wind>
 <speed value="7.7" unit="m/s" name="Moderate breeze"/>
 <gusts/>
 <direction value="170" code="S" name="South"/>
</wind>
<clouds value="40" name="scattered clouds"/>
<visibility value="10000"/>
<precipitation mode="no"/>
<weather number="802" value="scattered clouds" icon="03d"/>
<lastupdate value="2019-10-18T13:00:35"/>
</current>
```

Abbildung 4.2 Wetterdaten für Hamburg im XML-Format [OpenWeather 2019]

```
{
   "coord":{
      "lon":10,
      "lat":53.55
   },
   "weather":[
      {
         "id":800,
         "main":"Clear",
         "description":"clear sky",
         "icon":"01d"
      }
   ],
   "base":"stations",
   "main":{
      "temp":282.14,
      "pressure":1022,
      "humidity":66,
      "temp_min":279.82,
      "temp_max":284.82
   },
   "visibility":10000,
   "wind":{
      "speed":1
   },
   "clouds":{
      "all":0
   },
   "dt":1570439824,
   "sys":{
      "type":1,
      "id":1263,
      "message":0.0079,
      "country":"DE",
      "sunrise":1570426287,
      "sunset":1570466651
   },
   "timezone":7200,
   "id":2911298,
   "name":"Hamburg",
   "cod":200
}
```


Abbildung 4.3 Wetterdaten für Hamburg im JSON-Format [OpenWeather 2019]

4.3 Konzeption des ETL Prozesses

Der Reihe nach erfolgt hier die Konzeption des ETL Prozesses. In Kapitel 4.3.1 wird zunächst die Extraktion gezeigt, gefolgt von der Transformation in Kapitel 4.3.2, ehe Kapitel 4.3.3 das finale Laden der transformierten Daten erläutert.

4.3.1 Extraktion

Am Anfang des ETL-Prozesses, der Extraktion, werden alle benötigten Daten aus den zur Verfügung stehenden Quellen wie Datenbanken oder Dateien extrahiert bzw. in das Datenintegrationstool importiert.

Zur Verarbeitung der Daten müssen zunächst die beiden XLSX-Dateien „Import_Volvox.xlsx“ und „Artikel.xlsx“. importiert werden. Dieser Schritt ist in diesem Fall recht trivial und bedarf daher keiner größeren Konzeption.

4.3.2 Transformation

Nach Extraktion der Daten müssen diese transformiert werden.
Dabei sollten zunächst nicht benötigte Datensätze und Attribute herausgefiltert werden, sodass nur noch mit den relevanten Daten weitergearbeitet wird.
Anschließend werden die Daten in den Attributen so manipuliert, dass sie sich danach entweder schon im gewünschten Zielformat befinden oder aber, dass sie mit anderen Daten verknüpft werden können. Aus manchen Attributen müssen ggf. neue Daten berechnet werden. Des Weiteren ist es manchmal nötig, einem Datensatz neue weitere Attribute hinzuzufügen. Im letzten Schritt der Transformation werden die Daten dann beispielsweise in eine Datei exportiert.

Zunächst werden dazu aus der „Artikel.xslx“ alle Datensätze gefiltert werden, die zum Artikel „Espressivo Lehmfarbe“ gehören.
Das nächste Teilziel ist die Extraktion von Farbton und Gebindegröße aus der Spalte „Bezeichnung2“. In dieser Spalte befinden sich Strings, die neben diesen beiden Informationen auch Daten wie Preisgruppe (Buchstaben A, B, C, D und E), Verkaufseinheit und Palettengröße enthalten. Zur Extraktion der relevanten Daten eignen sich daher am besten String-Replace Funktionen, die mithilfe von regulären Ausdrücken nicht benötigte Daten aus den Strings herausfiltern sowie die relevanten Daten im einheitlichen Format darstellen.
So können einerseits die Preisgruppe und andererseits Farbton und Gebindegröße extrahiert werden.
Am Ende der String Replace Schritte soll im String daher der Farbton, gefolgt von einem Semikolon als Separator und zuletzt die Gebindegröße stehen.
Danach ist es dann möglich, Farbton und Gebindegröße mit einem String-Splitter am Semikolon in zwei Spalten aufzuteilen.
Des Weiteren können durch Map-Funktionen die Margen und Einkaufspreise anhand von Preisgruppen und Gebindegrößen zugeordnet und somit auch Verkaufspreise berechnet werden.
Es folgt eine Sortierung der Variationen nach Gebindegröße und Farbton, woraufhin mit einer Zählvariable und simplen String Verkettungen die interne Artikelnummer „ordnernumber“ des Betriebs „Ihr Farbraum Metzler & Block“ generiert wird.
„ordernumber“ muss beim ersten Datensatz auf „VVX0001“, d.h. den ersten Artikel von Volvox (VVX) gesetzt werden, alle anderen Datensätze erhalten ihre

Artikelnummer durch Komposition des ersten Strings, also „VVX0001“, mit einem Punkt sowie einer fortlaufenden ganzzahligen Nummer, beginnend mit „1“. Der erste Datensatz hat also hinterher den „ordernumber“ Wert „VVX0001“, der zweite „VVX0001.1“, der dritte „VVX0001.2“ usw.
Darüber hinaus wird der Wert „Kind“ gesetzt. Dieser sagt nur aus, ob es sich beim Artikel bzw. Datensatz um einen „parent“ (1), d.h. den Stammartikel oder um ein „child“ (2) also eine Variation bzw. einen Artikel handelt, der vom parent erbt. Der erste Datensatz der ursprünglichen Tabelle erhält hier also den „Kind“ Wert „1“, alle anderen den Wert „2“.
Nun muss das kartesische Produkt aus der fortlaufend manipulierten Artikeltabelle und der „Import Volvox“ Tabelle gebildet werden, damit am Ende jede Variation die benötigten Texte und Einheitsdaten enthält und die Transformation abgeschlossen ist.
Schlussendlich wird die finale Tabelle dann in eine CSV-Datei exportiert.

4.3.3 Laden

Nach erfolgreicher Extraktion und Transformation müssen die gewonnenen Daten ins Zielsystem geladen werden. Je nach Zielsystem unterscheidet sich hier die Vorgehensweise, manche Systeme bieten beispielsweise bereits interne Funktionen an, die das Laden von Daten erleichtern.

Die aus der Transformation resultierende CSV-Datei wird mit einer internen Shopware Funktion importiert. Anschließend stehen die Artikel im Shop zur Verfügung.
Auf diesen trivialen Schritt wird hier und auch bei der Umsetzung nicht näher eingegangen, da er im vorliegenden Praxisbeispiel keinen Wert für die Evaluation der beiden Tools hat.

5 Umsetzung

In diesem Kapitel geht es um die exemplarische Umsetzung der Konzeption mithilfe von Talend und Pentaho und den Daten des Betriebs Ihr Farbraum Metzler & Block sowie dessen Lieferanten. Zudem wird hier auch die Extraktion der Wetterdaten kurz beschrieben.
Zunächst werden dabei in 5.1 Extraktion und Transformation am Beispiel Pentaho demonstriert, ehe die Umsetzung dieser beiden ETL Schritte in 5.2 auch für Talend gezeigt wird. Wie schon kurz zuvor in Kapitel 4.3.3 erwähnt, ist der dritte Teil des ETL Prozesses, das „Laden“, bei dieser Umsetzung nicht relevant und wird hier daher auch bei keinem der beiden Tools beschrieben.

5.1 Umsetzung in Pentaho

In den folgenden Unterkapiteln wird die Extraktion und Transformation der Daten in Pentaho vorgestellt. Die Extraktion geht dabei auf alle vier Dateiformate ein. Die Transformation hingegen zeigt lediglich den Transformationsprozess der Artikeldaten und wird zum Ende hin stichpunktartig dargestellt, um einen kompakteren Überblick zu gewähren.

5.1.1 Extraktion

In Pentaho können XLSX-Dateien links unter „Design“ über ein „Microsoft Excel Input“ importiert werden. Diese Funktion erlaubt neben der Auswahl der Quelldatei die Angabe vieler weiterer Informationen, Metadaten und Eigenschaften, die beim Import berücksichtigt werden sollen. Dazu gehört beispielsweise die Selektion von benötigten Felden, also Spalten, samt Typzuordnung wie String, Integer etc. und Länge des Feldes.

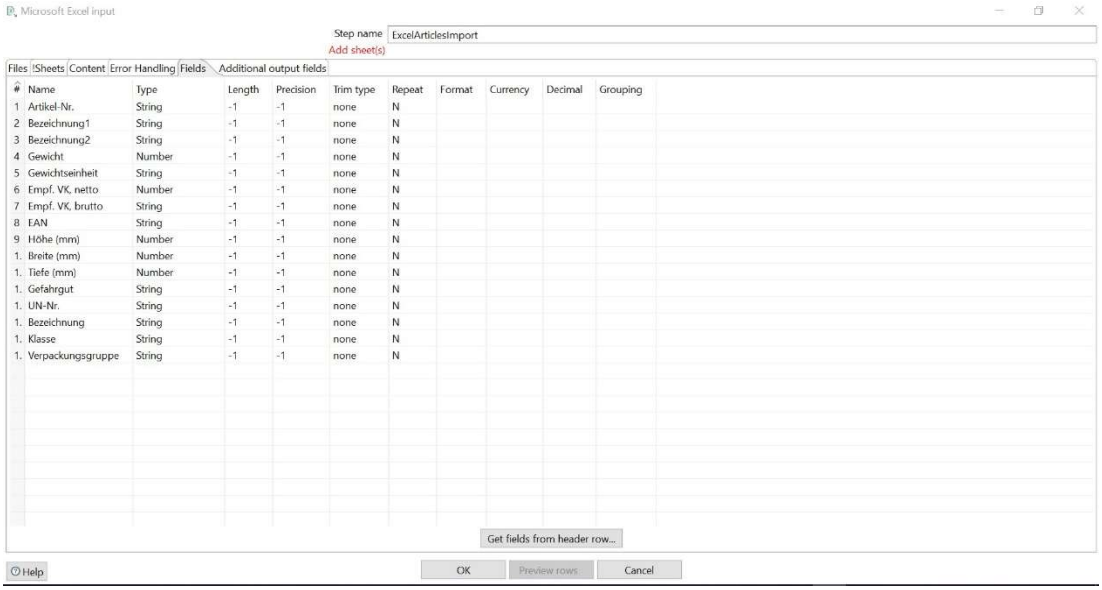
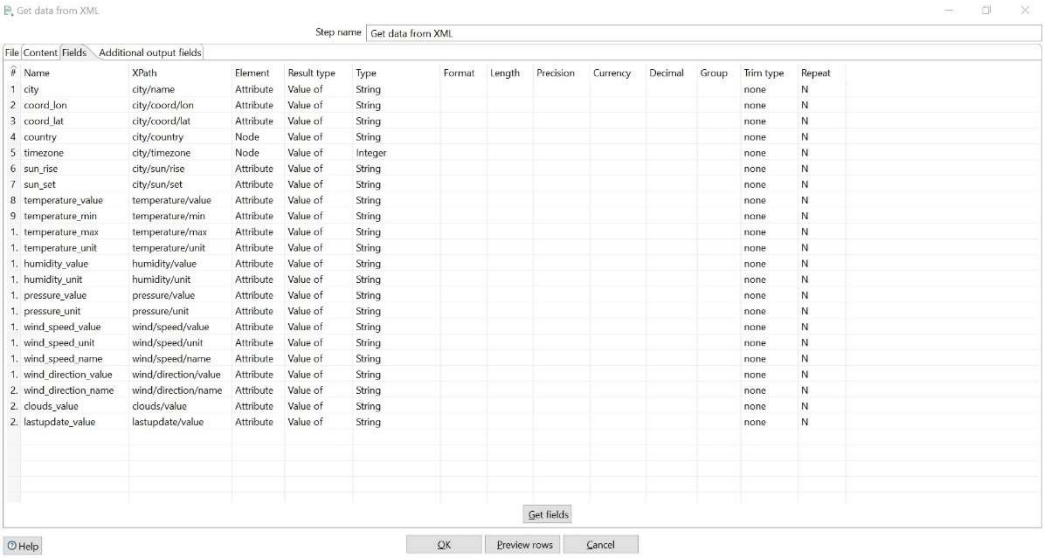


Abbildung 5.1 Tab „Fields“ in einer „Microsoft Excel input“ Transformation (Pentaho)

Über zwei dieser „Microsoft Excel input“ Schritte werden die beiden in Kapitel 4.2 vorgestellten XLSX-Dateien importiert, bzw. die Daten aus ihnen extrahiert.
Bei Import- sowie Export-Schritten in Pentaho können in den Dateipfaden auch Variablen wie „${Internal.Transformation.Filename.Directory}“ angegeben werden. Diese Variable enthält beispielsweise immer den aktuellen Dateipfad, in dem sich die Transformation an sich, also die „.ktr“-Datei, gerade befindet.
Daher wurde diese Variable auch bei jeglichen Import- und Export-Schritten in jeder implementierten Transformation benutzt, damit eine Ausführung auch noch möglich ist, wenn sie in einen anderen Ordner bzw. auf ein neues System kopiert wird. Eine manuelle Anpassung der Dateipfade entfällt somit, es müssen sich lediglich immer alle benötigten Import-Dateien und die eigentliche Transformation im selben Verzeichnis befinden.

XML-Dateien werden in Pentaho mit der „Get Data from XML“ Funktion importiert. Neben der Quelldatei kann hier im Reiter „content“ auch der „Loop XPath“ angegeben werden, also der Pfad in der XML-Datei, über den die Schleife läuft, wenn mehrere Datensätze mit gleichen Attributen vorhanden sind.
Im Reiter „Fields“ muss dann der XPATH angegeben werden, um auf die einzelnen Attribute zuzugreifen. Der XPATH erhält dabei durch Slashes („/“) eine Struktur, ähnlich wie Dateipfade in UNIX.



#	Name	XPath	Element	Result type	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	city	city/name	Attribute	Value of	String							none	N
2	coord_lon	city/coord/lon	Attribute	Value of	String							none	N
3	coord_lat	city/coord/lat	Attribute	Value of	String							none	N
4	country	city/country	Node	Value of	String							none	N
5	timezone	city/timezone	Node	Value of	Integer							none	N
6	sun_rise	city/sun/rise	Attribute	Value of	String							none	N
7	sun_set	city/sun/set	Attribute	Value of	String							none	N
8	temperature_value	temperature/value	Attribute	Value of	String							none	N
9	temperature_min	temperature/min	Attribute	Value of	String							none	N
1.	temperature_max	temperature/max	Attribute	Value of	String							none	N
1.	temperature_unit	temperature/unit	Attribute	Value of	String							none	N
1.	humidity_value	humidity/value	Attribute	Value of	String							none	N
1.	humidity_unit	humidity/unit	Attribute	Value of	String							none	N
1.	pressure_value	pressure/value	Attribute	Value of	String							none	N
1.	pressure_unit	pressure/unit	Attribute	Value of	String							none	N
1.	wind_speed_value	wind/speed/value	Attribute	Value of	String							none	N
1.	wind_speed_unit	wind/speed/unit	Attribute	Value of	String							none	N
1.	wind_speed_name	wind/speed/name	Attribute	Value of	String							none	N
1.	wind_direction_value	wind/direction/value	Attribute	Value of	String							none	N
2.	wind_direction_name	wind/direction/name	Attribute	Value of	String							none	N
2.	clouds_value	clouds/value	Attribute	Value of	String							none	N
2.	lastupdate_value	lastupdate/value	Attribute	Value of	String							none	N

Abbildung 5.2 Tab „Fields“ in der „Get Data from XML“ Transformation (Pentaho)

Der Import von JSON-Dateien erfolgt in Pentaho über das „JSON input“. Wie auch bei XML-Dateien muss hier im Tab „Fields“ der „Path“ also Pfad der Attribute definiert werden. Dieser beginnt mit einem Dollar Zeichen („$“), gefolgt von zwei Punkten („..“) und dem Namen des Attributes, sofern dies eindeutig ist. Alternativ kann der Pfad des Attributs statt mit zwei Punkten und dem Namen des eindeutigen Attributs auch durch Angabe des gesamten Pfads beschrieben werden, wobei ein einfacher Punkt („.“) zwischen den hierarchischen Bezeichnern steht.
Mit zwei eckigen Klammern („[]“) sowie einem Stern („*“) bzw. einer Zahl (z.B. „0“) zwischen beiden Klammern kann auf eine komplette Liste bzw. ein einzelnes Element der Liste in einer JSON-Datei zugegriffen werden.

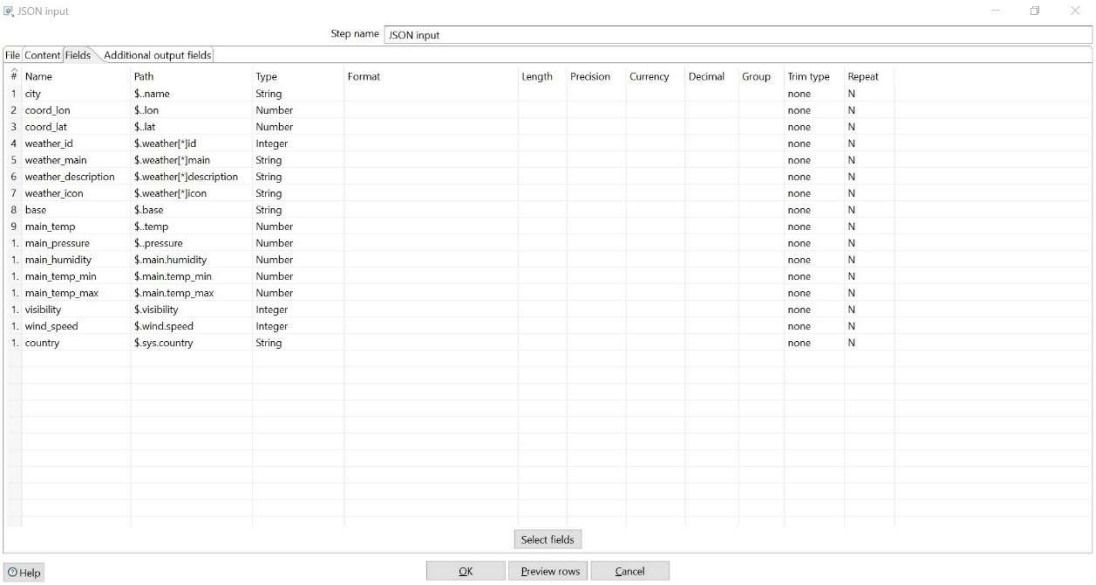


Abbildung 5.3 Tab „Fields“ in der „JSON input“ Transformation (Pentaho)

CSV-Dateien importiert Pentaho mithilfe des „CSV file input“. Hier kann neben Quelldatei bspw. der Delimiter festgelegt werden, in diesem Fall das Semikolon „;“. Zudem können natürlich die Felder samt Typzuordnung, Format und weiteren Eigenschaften bestimmt werden.

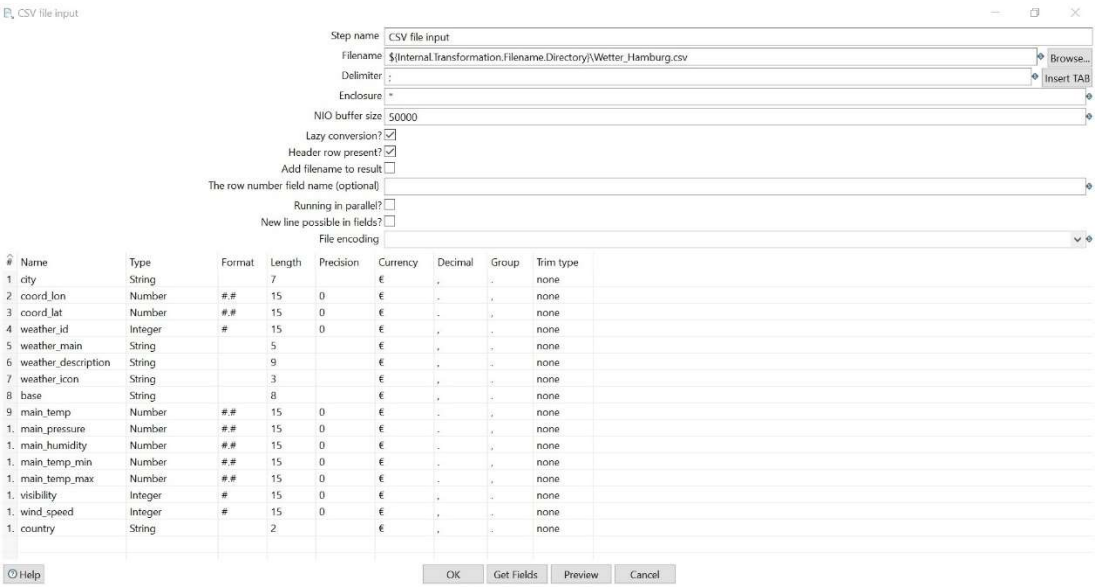


Abbildung 5.4 Screenshot der „CSV file input“ Transformation (Pentaho)

5.1.2 Transformation

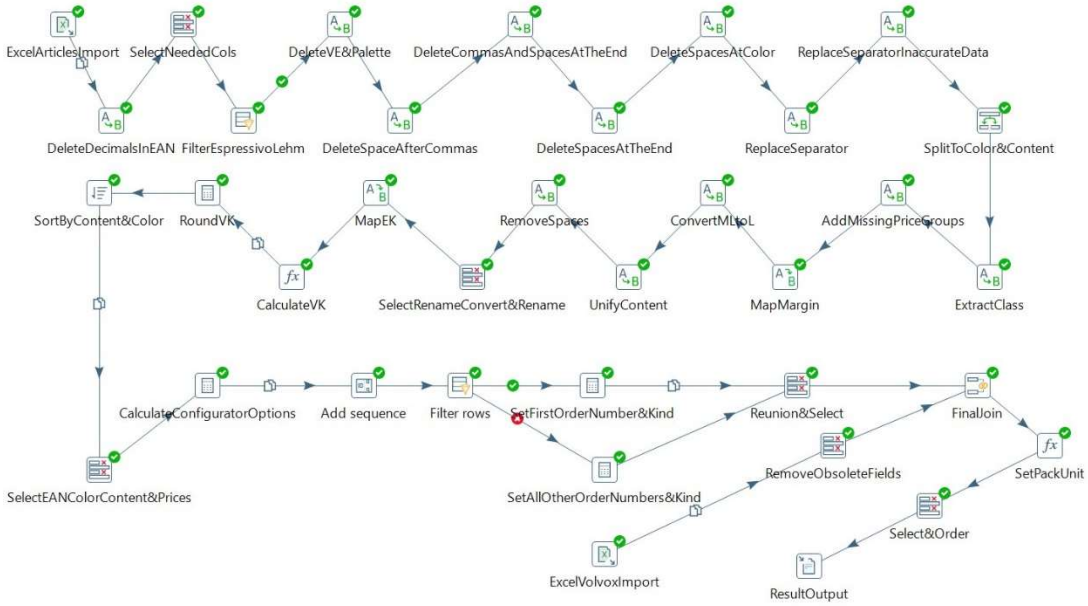


Abbildung 5.5 Designer Ansicht der gesamten Volvox Transformation in Pentaho

Nachdem die Daten in Pentaho geladen wurden, folgt zunächst eine Vielzahl an Schritten zur Gewinnung von Informationen aus der „Artikel.xlsx“ Datei, beginnend mit dem „DeleteDecimalsInEAN“ Schritt. Die „Import_Volvox.xlsx“ Datei wird erst in den letzten Zügen der gesamten Transformation in den Prozess aufgenommen und somit später eingebunden.
Der „DeleteDecimalsInEAN“ Schritt ist eine „String replace“ Transformation.
„String replace“ hilft bei der Durchsuchung und Manipulation von Strings. Man kann dabei entweder eine einfache Suche nach Substrings durchführen und diese ggf. ersetzen oder man führt das String replace Verfahren mittels Regulärer Ausdrücke durch, was komplexer ist, aber dafür auch wesentlich mehr Möglichkeiten bietet, wie die gleichzeitige Manipulation mehrerer Substrings in einem String.
Mit diesem ersten „String replace“ Schritt werden nun EAN Werte korrigiert. Dies war nötig, da Pentaho aus unerklärlichen Gründen einigen EAN Nummern ein Komma, gefolgt von einer „0“ anhängte, obwohl alle Datensätze den gleichen Import Prozess durchlaufen und die EAN Zellen in der .xlsx Datei weder Nachkommastellen enthalten noch verschiedene Zellformatierungen enthalten.
Nach einer „Select Values“ und einer „Filter Rows“ Transformation zum Selektieren benötigter Spalten sowie zum Filtern nach dem Artikel „Espressivo Lehmfarbe“ folgt eine Reihe von „String Replace“ Schritten, um Farbton und Gebinde ins korrekte

String Format zu transformieren und für die „Split Fields“ Transformation vorzubereiten. „Split Fields“ teilt nämlich beide Daten in separate Spalten auf. In vielen „String Replace“ Funktionen werden auch Inkonsistenzen behoben, die leider in der von Volvox zur Verfügung gestellten Datei auftraten, also beispielsweise fehlende Kommata, unterschiedlicher Aufbau der Strings etc.
Die Transformation wird dann folgendermaßen weitergeführt, wobei triviale Schritte wie „Select Values“ nicht nochmal aufgeführt werden:

- „String Replace“ und „Value Mapper“ Schritte: Extraktion der Preisgruppe, Konvertierung von Milliliter in Liter, Zuweisung der Margen und Einkaufspreise anhand von Preisgruppe und Gebindegröße (Margen und Preise wurden in dieser Arbeit mit Fantasiewerten belegt, da dies firmenbezogene Daten sind, die der Geheimhaltung unterliegen).
- „Formula“ und „Calculator“ zur Berechnung und Rundung der Verkaufspreise
- „Sort Rows“ zur Sortierung der Variationen nach Gebinde und Farbton
- „Calculator“ Transformation zur Generierung der „configuratorOptions“. Diese Spalte wird später im Shop benötigt, um auf der Website Dropdown Menüs zu befüllen, mit denen der Kunde Farbton und Gebindegröße auswählen kann.
- „Add Sequence“ und „Filter Rows“ Schritte zur Bestimmung der Werte „Kind“ und „ordernumber“
- „Join Rows (cartesian product)“ um das kartesische Produkt aus der manipulierten Artikelliste und der „Import_Volvox“ Tabelle zu generieren
- Wieder eine „Formula“ Funktion, diesmal um „packunit“, also die Verpackung je nach Gebindegröße festzulegen: „Dose“ bei höchstens einem Liter, „Eimer“ bei mehr als einem Liter
- „Text file output“ Schritt, bei dem die finale Tabelle in eine CSV-Datei geschrieben wird

5.2 Umsetzung in Talend

Kapitel 5.2.1 und 5.2.2 veranschaulichen Extraktion und Transformation des ETL Prozesses in Talend. Auch in diesem Fall werden wie bei der Umsetzung in Pentaho lediglich die Artikeldaten behandelt und nicht die Wetterdaten.

5.2.1 Extraktion

Die Extraktion von Daten aus einer XLSX-Datei erfolgt in Talend zunächst über ein „tFileInputExcel“. Dabei ist der Assistent zum Erzeugen bzw. Importieren einer Excel

Datei hilfreich, den man links in der „Ablage“ unter „Meta-Daten“ mit einem Rechtsklick auf „Excel Datei“ -> „Excel Datei erstellen“ aufrufen kann. Mithilfe des Assistenten können nun die Quelldatei ausgewählt und ähnlich wie in Pentaho Parameter des Imports festgelegt werden, also bspw. welche Tabellenblätter und Spalten relevant sind und welche Datentypen in den einzelnen Spalten vorliegen.
Wie bei der Extraktion in Pentaho werden logischerweise auch hier die beiden Dateien „Artikel.xlsx“ und „Import_Volvox.xlsx“ importiert.

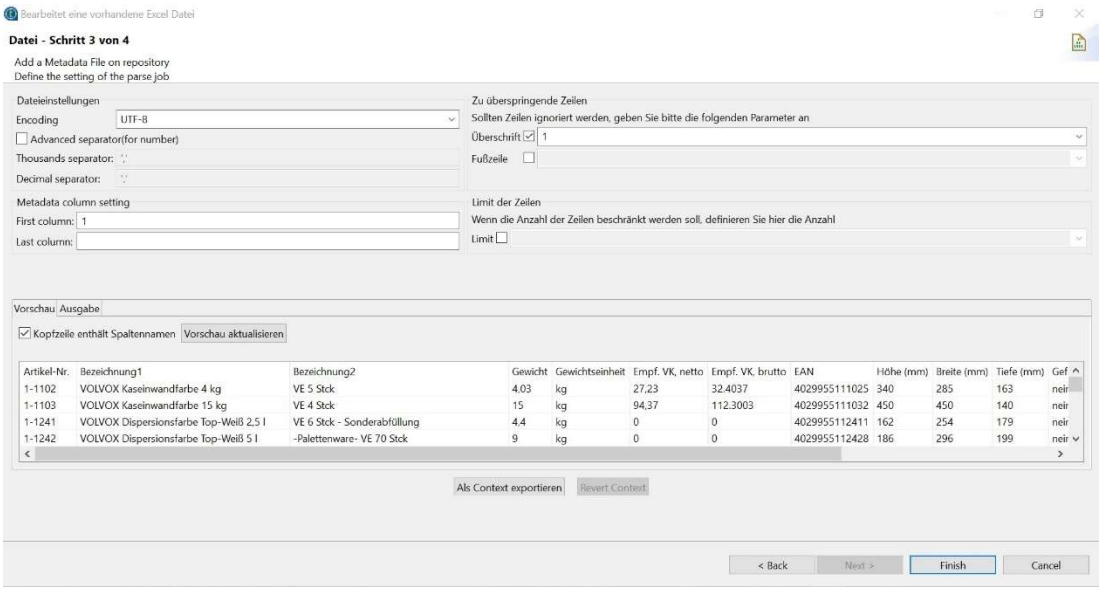


Abbildung 5.6 Teilschritt beim Import einer XLSX-Datei in Talend

XML-, JSON- sowie CSV-Dateien können in Talend relativ einfach mit dem Assistenten importiert werden. Dazu wählt man unter „Meta-Daten“ -> „XML Datei“/“File Json“/“File delimited“ mit der rechten Maustaste „Erstelle Datei XML“/“Create JSON Schema“/“Erstelle File delimited“ aus, um den Assistenten zu starten. Anschließend hilft dieser bei Auswahl der Quelldatei sowie der gewünschten Felder bzw. Attribute. Bei XML- und JSON- Dateien muss daneben wie bei Pentaho noch der Loop Path angegeben werden.

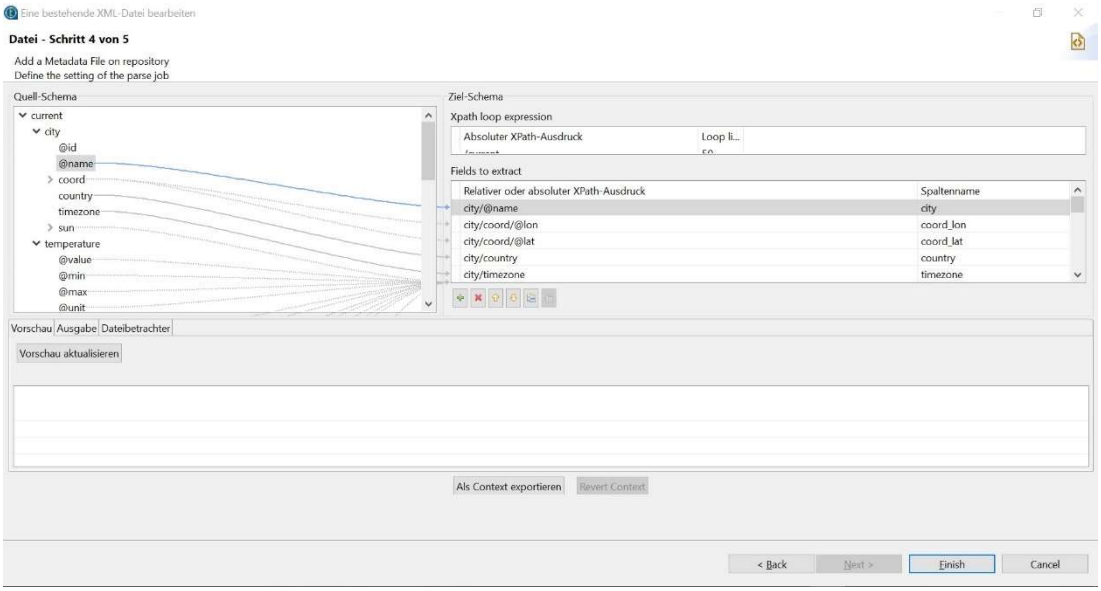


Abbildung 5.7 Teilschritt beim Import einer XML-Datei in Talend

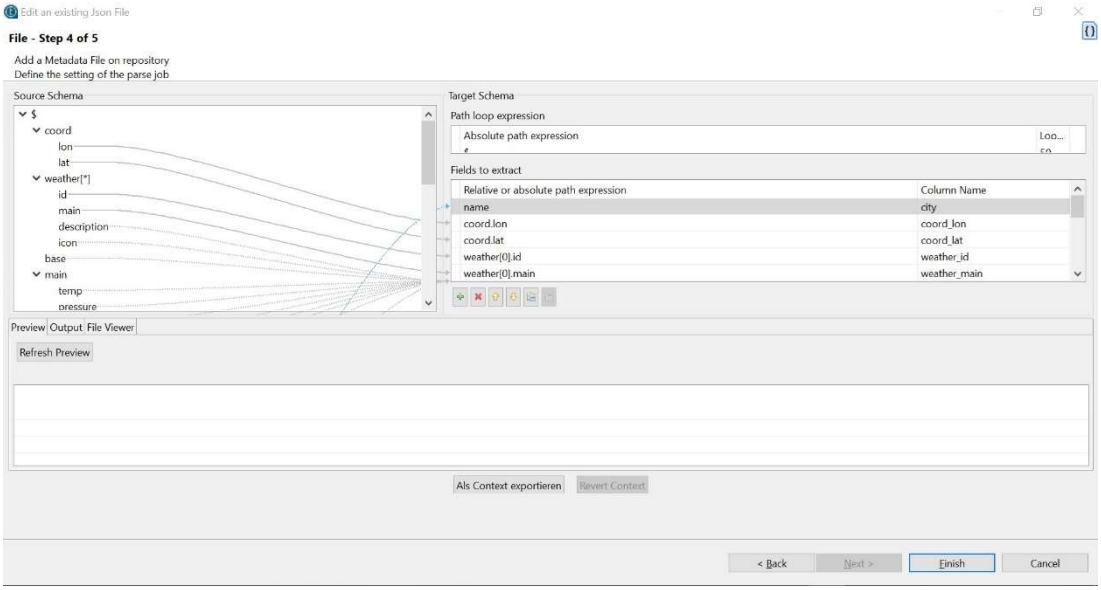


Abbildung 5.8 Teilschritt beim Import einer JSON-Datei in Talend

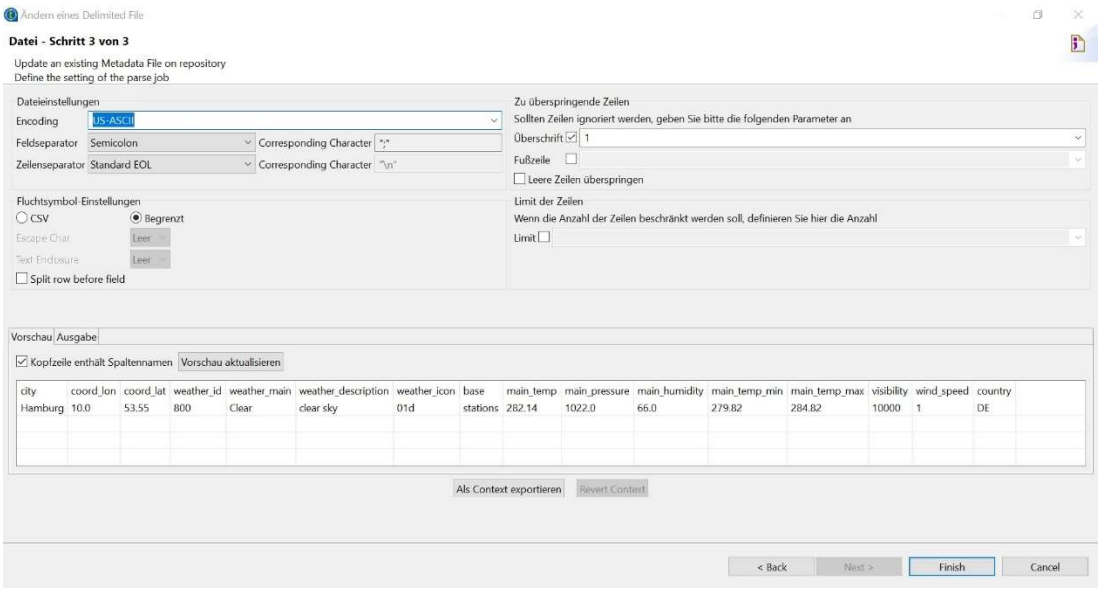


Abbildung 5.9 Teilschritt beim Import einer CSV-Datei („File delimited) in Talend

5.2.2 Transformation

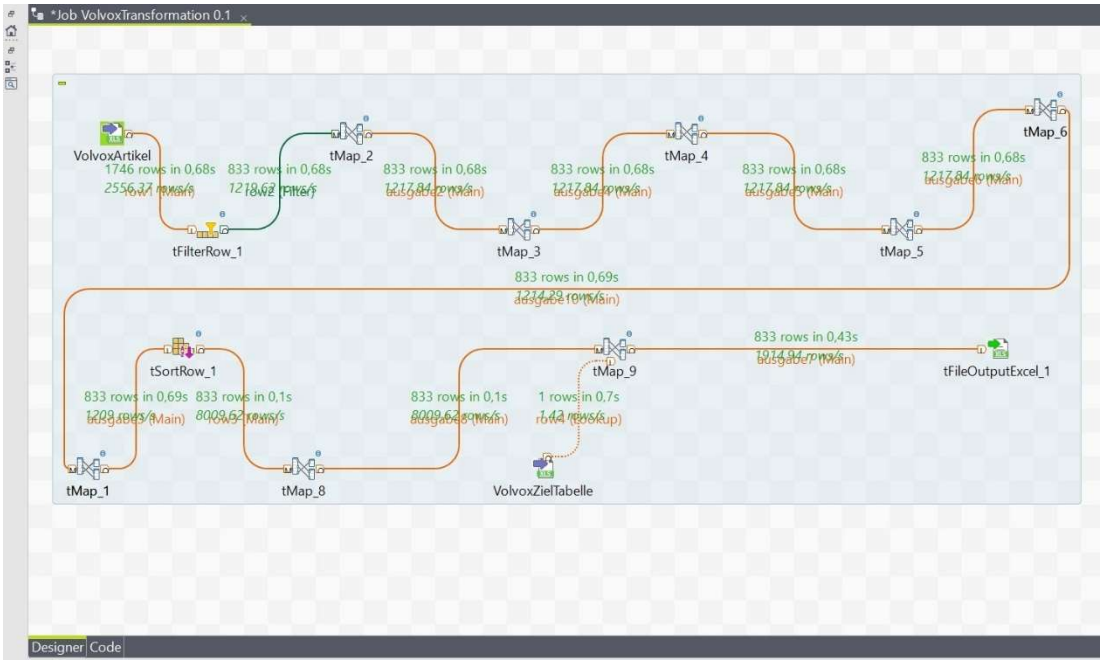


Abbildung 5.10 Designer Ansicht der gesamten Volvox Transformation in Talend

Im ersten Schritt der Transformation in Talend werden zunächst mit der „tFilterRow“ Funktion die Zeilen herausgesucht, die zum Artikel „Espressivo Lehmfarbe“ gehören, alle anderen werden herausgefiltert. „tFilterRow“ bietet die Möglichkeit, Spalten auf verschiedene Bedingungen, wie Gleichheit und Ungleichheit zu prüfen und danach zu filtern. Mehrere Bedingungen können durch das logische „UND“ bzw. das logische „ODER“ verknüpft werden. In diesem Beispiel wird lediglich eine Bedingung benötigt, die die Spalte „Bezeichnung1“ auf Gleichheit mit dem String „VOLVOX Espressivo Lehmfarbe“ überprüft.
Es folgt eine „tMap“ Transformation. Dies ist eine sehr vielseitige und mächtige Transformation, genau wie die „Map“ Funktion in der Programmiersprache Java, was kein Zufall ist, da Talend aus allen Transformationen im Hintergrund Java Quellcode erzeugt, der dann die eigentliche Transformation darstellt und für die Ausführung genutzt wird. Aus diesem Grund kann in Talend auch eigener Java Code geschrieben und genutzt werden, was in dieser Arbeit auch geschieht.
Durch den großen Funktionsumfang der „tMap“ Funktion, kommt die gesamte Transformation in Talend auch fast ausschließlich mit „tMap“ Schritten aus. Daher folgt wie auch bei der Transformation in Pentaho nun eine stichpunktartige Auflistung der weiteren Etappen:

- „tMap_2“: Selektion der benötigten Spalten, Entfernung von Palettengröße und Verkaufseinheit im String „Bezeichnung2“ durch „String.replaceAll()“
- „tMap_3“, „tMap_4“: Leerzeichen und Kommata im String ersetzen
- „tMap_5“: Komma Trennzeichen durch Semikolons ersetzen, inkonsistente Datensätze bereinigen
- „tMap_6“: Aufteilung von Farbton und Gebindegröße in zwei Spalten am Trennzeichen Semikolon durch Verwendung von „String.split()“
- „tMap_1“: Konvertierung von Milliliter in Liter, weitere Bereinigung von Inkonsistenzen, Extraktion der Preisgruppe
- „tSortRow_1“: Sortierung der Zeilen bzw. in diesem Fall Variationen nach Farbe und Gebinde
- „tMap_8“: Bestimmung der „configuratorOptions“ (String aus u.a. Farbton und Gebindegröße), Generierung von „ordernumber“ und „Kind“ mithilfe eines durch „Numeric.sequence()“ erzeugten counters, Zuweisung von Marge und Einkaufspreis mit der „String.equals()“ Funktion, Berechnung des Verkaufspreises und Bestimmung (auch hier wurden wie in Pentaho Preise und Margen zum Zweck der Geheimhaltung mit Beispielwerten belegt).

- „tMap_9“: Bildung des Kreuzprodukts aus der transformierten Artikelliste und der bereits importierten „Import_Volvox“ Tabelle sowie Anordnung der Spalten
- „tFileOutputExcel“: Export der finalen Tabelle in eine CSV-Datei

In „tMap_8“ wurden neben den Standardfunktionen von Talend auch die beiden statischen Funktionen „RoundDouble.roundDouble()“ und „StringWithCommaToDouble.stringWithCommaToDouble()“ genutzt, um Verkaufspreise zu kalkulieren und „packunit“ zu definieren. Diese mussten selbst implementiert werden, da im Funktionsumfang von Talend weder eine Funktion zur Rundung von Double Werten enthalten ist, noch eine zur Konvertierung von Dezimalzahlen im String Format in einen Double Wert.
„RoundDouble.roundDouble()“ bekommt einen Double Wert d und einen Integer p als Parameter übergeben und gibt anschließend den Wert d auf p Nachkommastellen gerundet aus.
„StringWithCommaToDouble.stringWithCommaToDouble()“ erwartet als Eingabewert hingegen einen String „number“, der eine Dezimalzahl mit Dezimalkomma enthält. Zurückgegeben wird dann ein Double mit dem Dezimalwert des Strings „number“.

6 Bewertung

Kapitel 6 beschreibt die Bewertung, also die finale Evaluation. Dabei wird zunächst in 6.1 bewertet, inwiefern die Quelldateien geeignet sind, um mithilfe von Pentaho und Talend Daten aus ihnen zu extrahieren, zu transformieren und zu laden. Anschließend erfolgt in 6.2 die tatsächliche Bewertung der beiden Tools.

6.1 Bewertung der Dateien

In den folgenden beiden Unterkapiteln werden einerseits die „Artikel.xlsx“ Datei und andererseits die drei Dateien mit Wetterdaten auf Ihre Eignung zur Evaluierung beider Tools bewertet.
Die eigens erstellte Datei „Import_Volvox.xlsx“ wird nicht bewertet, da sie nicht von fremden Quellen bezogen wurde und ohnehin nur eine einzige Zeile besitzt, die hauptsächlich Beschreibungstexte enthält.

6.1.1 Bewertung der Datei „Artikel.xlsx“

Die Datei „Artikel.xlsx“ mit Produkten von Volvox wurde wie bereits erwähnt von der „Ecotec Naturfarben GmbH“ zur Verfügung gestellt. Sie enthält wie auch in den Anforderungen (Kapitel 3.2.1) verlangt Artikel/-daten, die in den Online Shop aufgenommen werden sollen
Die vorliegenden Artikeldaten wie Produktname, Farbton und Gebindegröße zu Volvox Produkten wurden importiert und transformiert.
Leider waren die Daten nicht optimal zur Transformation geeignet. Produktname und EAN Code waren separat in jeweils einer Spalte vorhanden und konnten daher sehr einfach verarbeitet werden.
Farbton, Gebindegröße und Preisgruppe hingegen sind hier alle mit den nicht relevanten Daten „Palettengröße“ und „Verkaufseinheit“ in einer Spalte

„Bezeichnung2“ zusammengefasst, was aus vier Gründen unabhängig vom eingesetzten Tool zur automatisierten Datenintegration unvorteilhaft ist:

(I) Das Auslesen bzw. Verarbeiten der Daten ist wesentlich schwieriger. Würden die Daten in separaten Spalten vorliegen, könnte direkt mit den Attributen gearbeitet werden. So müssen die Attribute aber erstmal aus dem gesamten String extrahiert werden
(II) Die einzelnen Daten müssen mit Trennzeichen wie in diesem Fall Kommata und Leerzeichen voneinander getrennt werden, damit eine Unterscheidung der Attribute möglich ist
(III) Punkt (II) hat zur Folge, dass schneller Fehler und Inkonsistenzen wie fehlende Kommata oder Leerzeichen entstehen
(IV) Die Extraktion der Attribute wird zusätzlich erschwert, da Kommata als Separator der Attribute und zugleich auch bei der Gebindegröße als Dezimalkomma genutzt werden

Tatsächlich enthielt die Datei „Artikel.xlsx“ einige Inkonsistenzen, die sich im Fehlen von Kommata bzw. Leerzeichen als auch in Preisgruppen widerspiegelten. Zudem waren die Werte Verkaufseinheit und Palettengröße auch nicht immer angegeben. Diese Inkonsistenzen (III) in Verbindung mit den drei anderen genannten Problemen (I), (II) und (IV) erschwerten dementsprechend den Import bei beiden Tools. Der Vorteil der Inkonsistenzen ist hingegen, dass dadurch die Tools besser evaluiert werden können.

6.1.2 Bewertung der Dateien mit Wetterdaten

Die Dateien „Wetter_Hamburg.json“ und „Wetter_Hamburg.xml“ wurden mithilfe der Web API von OpenWeatherMap [OpenWeather 2019] erzeugt. Beide erfüllen die Anforderungen, da sie konsistent sind, sich an RFC 8259 [Bray 2018] bzw. RFC 3470 [Hollenbeck u.a. 2003] halten und Beispieldaten im JSON- bzw. im XML-Format bereitstellen.
Die Datei „Wetter_Hamburg.csv“ wurde wie bereits erwähnt durch die Transformation der Wetterdaten generiert. Daher erfüllt sie auch alle Anforderungen an die CSV Datei, wie auch RFC 4180 [Shafranovich 2005].

6.2 Bewertung der Tools

In den nächsten beiden Kapiteln werden die beiden Tools Pentaho und Talend bewertet, indem jeweils überprüft wird, inwieweit die funktionalen und

nichtfunktionalen Anforderungen erfüllt werden. Ein finaler Vergleich beider Tools erfolgt dann im Fazit.

6.2.1 Bewertung zum Tool Pentaho

In diesem Abschnitt wird das Tool Pentaho bewertet. Dabei erfolgt zunächst eine methodische Bewertung. Anschließend wird beurteilt, wie gut Pentaho im expliziten Fallbeispiel geeignet ist.

6.2.1.1 Methodik

Das Einlesen von XLSX- (FA1), CSV- (FA2), XML- (FA3) sowie von JSON-Dateien (FA4) funktionierte in Pentaho mit den jeweiligen Input-Transformationen und Assistenten sehr gut und einfach. Import und Manipulation der Daten war problemlos möglich. Lediglich nach der Extraktion der Daten aus einer der XLSX-Dateien trat wie bereits erwähnt ein nicht nachvollziehbarer kleiner Fehler auf, als manchen aber nicht allen EAN Codes eine Nachkommastelle angehängt wurde.
Darüber hinaus traten allerdings keine weiteren Probleme auf, also erfüllt Pentaho (FA1) bis (FA4).
Die Transformation der importierten Daten ins benötigte Zielformat konnte in Pentaho ebenfalls erfolgreich durchgeführt werden. Nach einer kurzen Eingewöhnungs- und Lernzeit lassen sich Daten mit den verschiedenen Funktionen bzw. Transformationen sehr gut manipulieren. Dies ist teilweise auch relativ einfach und selbsterklärend, wenn es bspw. um die Auswahl von Attributen, die Filterung und Sortierung von Datensätzen oder einfache Berechnungen geht. Die Manipulation von Strings ist in der Regel allerdings etwas komplexer, vor allem wenn mit regulären Ausdrücken gearbeitet wird. (FA5) ist somit erfüllt.
Das Laden bzw. Exportieren der transformierten Daten in eine CSV-Datei funktionierte unproblematisch und ziemlich selbsterklärend, d.h. Pentaho genügt auch hier der Anforderung (FA6).
Inkonsistente Datensätze wurden in Pentaho leider nicht automatisch bereinigt, hier musste händisch nachgeholfen werden. Inkonsistenzen wurden durch Betrachtung der Datensätze und durch fehlerhafte Datenmanipulationen in nachfolgenden Transformationsschritten identifiziert und daraufhin in zusätzlichen Schritten bereinigt.
Darüber hinaus baute Pentaho wie schon erwähnt selbst noch bei der Extraktion Inkonsistenzen ein, indem es Strings aus Ziffern teilweise trotzdem als Zahl behandelte und ihnen daher eine Nachkommastelle anhängte. Dies geschah allerdings nur bei einigen Strings und trat wiederum bei anderen nicht auf, obwohl sich diese Strings alle in derselben Spalte - folglich im selben Attribut - befanden

und die gleiche Formatierung aufwiesen. Hier erfüllt Pentaho die Anforderung (FA7) also nicht, bzw. nur auf einfachste Weise, wenn man die Inkonsistenzen manuell mit Pentaho behebt.
Wie bereits erwähnt gab es kleine Fehler beim Extrahieren der Daten. Ansonsten konnte der ETL Prozess mit Pentaho aber insgesamt korrekt und vollständig angelegt und ausgeführt werden, auch wenn dazu händisch Inkonsistenzen bereinigt werden mussten. Somit wird hier von Pentaho die erste nichtfunktionale Anforderung erfüllt (NFA1).
Pentaho ist auch für Einsteiger geeignet und mit ein wenig Zeit relativ leicht zu verstehen. Zumindest das Umstellen auf eine neue Import Datei sowie die Ausführung des ETL Prozesses ist auch für weniger erfahrene Nutzer durchführbar, weshalb Pentaho (NFA2) genügt.
Zur Überprüfung des Zeitverhaltens (NFA3) von Pentaho wurden einige Zeitmessungen durchgeführt. Diese fanden alle auf demselben Gerät statt (Laptop von Acer mit Windows 10 Pro, 64 Bit, 16 GB RAM, Intel Core i7-8550U -> 4 Kerne, 8 Threads @ 1,80GHz). Als Grundlage diente hier das Beispielprojekt bzw. der exemplarisch umgesetzte ETL Prozess. Bei umfangreicheren Prozessen mit größeren Daten und komplexeren Transformationen würden die Messungen natürlich andere Werte liefern.
Hier aber nun die Messergebnisse anhand des Beispielprojekts:

	1. Messung	2. Messung	3. Messung
Start des Tools[1]	----------	----------	----------
Erster Start[2]	2:32min	2:31min	/
Weiterer Start[3]	2:01min	2:02min	/
Ausführung[4]	----------	----------	----------
Erste Ausf.[5]	6.65s	5.03s	/
Weitere Ausf.[6]	3.46s	3.72s	3.33s

Tabelle 6.1 Zeitmessungen: Programmstart & Ausführung der Transformation (Pentaho)

Legende der Tabelle 6.1:
[1]Start des Programms inkl. Öffnung des Projekts
[2]Erster Start nach Hochfahren des Betriebssystems
[3]Weitere Starts (Pentaho wurde bereits einmal gestartet und wieder geschlossen)
[4]Komplette Ausführung des ETL-Prozesses
[5]Erste Ausführung direkt nach Start des Programms
[6]Weitere Ausführungen (ETL Prozess wurde nach Programmstart bereits mindestens einmal ausgeführt)

Pentaho braucht zum Starten also ca. zwei bis zweieinhalb Minuten, je nachdem, ob es sich um den ersten Start nach dem Booten des Betriebssystems handelt oder nicht. Damit ist es deutlich langsamer als Alltagstools wie beispielweise Office Programme oder Webbrowser, liegt aber immer noch unter der gesetzten Grenze von vier Minuten. Auch die Ausführung des ETL Prozesses liegt mit ca. drei bis sechseinhalb Sekunden deutlich unter der angeforderten Maximaldauer von einer Minute. Somit gewährleistet Pentaho auch die dritte nichtfunktionale Anforderung (NFA3).
Pentaho ist auch für Neulinge geeignet, da es viele selbsterklärende Transformationen bereitstellt und diese übersichtlich darstellt. Auch die Anordnung und Verbindung der Transformationen durch Drag & Drop ist sehr benutzerfreundlich. Zusätzlich hilft der Assistent in Pentaho bei der Umsetzung der Transformationen. Der ETL Prozess in diesem Beispiel ist hingegen nicht unbedingt direkt und einfach für Neulinge zu verstehen, was vor allem an den regulären Ausdrücken liegt. Insgesamt erfüllt Pentaho die Anforderung (NFA4) somit nur teilweise.
Die soeben genannten Eigenschaften von Pentaho, also das Drag & Drop, der Assistent sowie die oft selbsterklärenden Transformationen machen es auch Nicht-ITlern einfach, zumindest erste simple ETL Prozesse selbst zu entwickeln. Komplexere Transformationen wie beispielsweise durch Verwendung von regulären Ausdrücken werden hingegen ohne Vorkenntnisse schwer für Nicht-ITler umzusetzen sein. Daher ist auch die Modifikation des vorhandenen ETL Prozesses für diese Menschen nur bedingt möglich. Die letzte Anforderung (NFA5) wird somit von Pentaho nur zum kleinen Teil erfüllt.

6.2.1.2. Fallstudie

Wie bereits beschrieben, erfüllt Pentaho (NFA1) und (NFA2). Der ETL Prozess verlief korrekt und ein Mitarbeiter konnte diesen auch ohne große Vorkenntnisse und Anleitung ausführen sowie eine neue Quelldatei auswählen. (NFA4) und (NFA5) wurden nur minimal erfüllt, keiner der Mitarbeiter konnte den gesamten Prozess verstehen oder gar anpassen. Da allerdings viele Transformationen bei Pentaho selbsterklärend sind und die Benutzeroberfläche intuitiv ist, konnten kleine simple Transformationen auch eigenständig von einem Mitarbeiter erstellt werden. Auch der Zeitaufwand blieb bei Pentaho im Rahmen (NFA3).
Somit ist Pentaho insgesamt sehr gut für den Betrieb geeignet, auch in Zukunft können die Mitarbeiter mithilfe dieses Tools neue Quelldateien transformieren und ggf. kleinere Änderungen selbst vornehmen.

6.2.2 Bewertung zum Tool Talend

Analog zur Bewertung bei Pentaho wird erfolgt auch hier eine separate Beurteilung des Tools Talend im Kontext der Methodik sowie der Fallstudie.

6.2.2.1. Methodik

Ebenso wie bei Pentaho war bei Talend der Import und die Manipulation von Daten aus XLSX-, CSV-, XML- und JSON-Dateien problemlos möglich. Auch hier half ein Assistent bei der Auswahl der Quelldatei, Attributen usw. Es benötigte lediglich ein wenig mehr Aufwand als bei Pentaho, da mit dem Assistenten zunächst Meta-Daten zu einem Dateityp festgelegt werden mussten, ehe die Input Transformation des jeweiligen Dateityps in den Job eingefügt werden konnte. Dafür traten beim Import aller Beispieldateien keine Fehler auf, es wurden also nicht wie bei Pentaho manchen EAN Codes Nachkommastellen angehängt. (FA1) bis (FA4) werden von Talend also erfüllt.
Auch in Talend konnten die Quelldaten erfolgreich ins gewünschte Zielformat transformiert werden. Insgesamt nahm die Programmierung des ETL Prozesses allerdings mehr Zeit in Anspruch als Pentaho. Dies lag insbesondere an der weniger intuitiven und gerade für Einsteiger komplexeren Benutzeroberfläche. Wie schon gesagt, mussten zunächst Meta-Daten festgelegt werden, um Dateien zu importieren. Außerdem funktionierte das Dag & Drop System, sowie das Verbinden von zwei Transformationen nicht so perfekt. Oftmals verschiebt man dadurch die falsche Transformation oder muss den Ausgabestream ändern bzw. umbenennen. Darüber hinaus war „tMap“ der am häufigsten genutzte Transformationsschritt in der Beispielanwendung, was die Mächtigkeit dieser Funktion beweist. Der Nachteil daran ist jedoch, dass hierbei meistens Java Funktionen genutzt werden müssen, was insbesondere für Nicht-ITler und Menschen ohne Java Vorkenntnisse schwierig zu realisieren ist, bzw. mehr Zeit zur Einarbeitung benötigt. Hier zeigt sich direkt eine weitere Eigenschaft von Talend, nämlich dass neue Java Funktionen auch selbst implementiert werden können, was das Tool einerseits wieder mächtiger, andererseits aber auch komplexer als Pentaho macht. Auch in der Beispielanwendung mussten zwei Funktionen selbst programmiert werden, da diese nicht wie bei Pentaho schon mit dem Tool geliefert wurden. Talend genügt also insgesamt (FA5), ist allerdings mächtiger und damit zugleich komplexer als Pentaho.
Das Exportieren der Daten in eine CSV-Datei verlief reibungslos, auch hier erfüllt Talend also die funktionale Anforderung (FA6).
Ebenso wie Pentaho bereinigt Talend inkonsistente Datensätze nicht automatisch. Fehler und Inkonsistenzen müssen also ebenso mit diesem Tool von Hand bereinigt werden. Dabei wurden analog zu Pentaho fehlerhafte Datensätze identifiziert und

mit zusätzlichen Transformationsschritten korrigiert. Im Gegensatz zu Pentaho baute Talend aber keine zusätzlichen Fehler in den Daten ein, erfüllt daher (FA7) etwas mehr als Pentaho, wenn auch nicht vollständig, weil auch hier keine automatische Bereinigung erfolgt.
Der gesamte ETL Prozess wurde von Talend vollständig durchlaufen und die resultierende Datei war korrekt, also ist auch hier die nichtfunktionale Anforderung (NFA1) erfüllt.
Wie bereits bei der Verifizierung von (FA5) angesprochen, ist Talend nicht so intuitiv und leicht zu bedienen wie Pentaho, vor allem für Einsteiger. Wenn in Zukunft eine andere Datei für den Import genutzt wird, muss höchstwahrscheinlich die Meta-Datei angepasst bzw. eine neue angelegt werden, was für einen Anfänger ohne Hilfe und Anleitung schwierig werden könnte. Den ETL Prozess könnte er allerdings auch ohne große Vorkenntnisse ausführen, weshalb Talend (NFA2) nur teilweise erfüllt.
Auch zur Überprüfung des Zeitverhaltens von Talend wurde ebenso wie bei der Verifizierung von (NFA3) bei Pentaho auf demselben PC die gleichen Zeitmessungen durchgeführt. Dazu wurde erneut das Beispielprojekt herangezogen, komplexere Projekte würden hier natürlich auch andere Ergebnisse liefern. Da bei Talend im Gegensatz zu Pentaho nicht automatisch das zuletzt geöffnete Projekt gestartet wird, sondern während des Starts das Projekt per Hand ausgewählt werden muss, wurde die Zeit während der Auswahl angehalten.
Die Zeitmessungen von Talend lieferten folgende Ergebnisse:

	1. Messung	2. Messung	3. Messung
Start des Tools[1]	----------	----------	----------
Erster Start[2]	1:43min	1:46min	/
Weiterer Start[3]	39.65s	40.60s	/
Ausführung[4]	----------	----------	----------
Erste Ausf.[5]	13.45s	14.52s	/
Weitere Ausf.[6]	6.15s	5.41s	4.96s

Tabelle 6.2 Zeitmessungen: Programmstart & Ausführung der Transformation (Talend)

Legende der Tabelle 6.2:
[1]Start des Programms inkl. Öffnung des Projekts
[2]Erster Start nach Hochfahren des Betriebssystems
[3]Weitere Starts (Talend wurde bereits einmal gestartet und wieder geschlossen)
[4]Komplette Ausführung des ETL-Prozesses
[5]Erste Ausführung direkt nach Start des Programms
[6]Weitere Ausführungen (ETL Prozess wurde nach Programmstart bereits mindestens einmal ausgeführt)

Der erste Start von Talend nach dem Booten des Betriebssystems dauert mit ca. einer Minute und 45 Sekunden noch relativ lange, alle weiteren Starts verlaufen mit ungefähr 40 Sekunden dafür aber relativ schnell. In beiden Fällen liegt Talend jedoch unter der Grenze von zwei Minuten. Die erste Ausführung des Beispielprojekts benötigt in Talend durchschnittlich ca. 14 Sekunden, während weitere Durchläufe im Schnitt bei ungefähr 5,5 Sekunden liegen. Die Durchführung der Transformation liegt also deutlich unter der geforderten Obergrenze von einer Minute, womit Talend auch (NFA3) genügt.
Wie schon bei der Überprüfung von (FA5) angesprochen, ist Talend für Neulinge eher ungeeignet, die Benutzeroberfläche ist nicht sehr intuitiv und es werden auch Java Kenntnisse benötigt, um Transformationen zu verstehen und zu programmieren. Daher ist der vorhandene ETL Prozess für Einsteiger eher schwer zu verstehen und noch schwerer zu modifizieren. Somit erfüllt Talend (NFA4) und (NFA 5) nicht.

6.2.2.2. Fallstudie

(NFA1) wird von Talend wie zuvor bereits gezeigt erfüllt. Allerdings ist Talend für Anfänger eher ungeeignet. (NFA2) wird nur teilweise erfüllt, (NFA4) und (NFA5) gar nicht. Dass Talend (NFA3) erfüllt und somit vom Zeitaufwand im Rahmen bleibt ist daher nicht mehr ausschlaggebend. Für den Betrieb „Ihr Farbraum Metzler & Block“ ist insbesondere wichtig, dass auch andere Mitarbeiter in Zukunft leicht neue Quelldateien mit aktualisierten Preisen importieren und den ETL Prozess ausführen können. Auch wäre es wünschenswert, dass die Mitarbeiter kleinere Anpassungen vornehmen können. Keiner der Mitarbeiter fand sich allerdings auf Anhieb in Talend zurecht. Die Ausführung des Prozesses gelang zwar, aber schon bei der Auswahl einer neuen Quelldatei traten Probleme auf. Zusammengefasst ist Talend also in dieser Fallstudie und für diesen Betrieb eher ungeeignet.

6.3 Fazit

Im nächsten Abschnitt wird ein Fazit gezogen und dabei nochmal ein abschließender Vergleich zwischen beiden Tools aufgezeigt. Auch hier wird erneut zwischen Methodik und Fallstudie unterschieden.

6.3.1 Methodik

Die vier Dateien waren gut geeignet, um die Anforderungen (FA1) bis (FA4) zu überprüfen. Pentaho und Talend erfüllen (FA1) bis (FA4). Mithilfe der XLSX-Datei

konnte dann in beiden Tools eine geeignete Transformation programmiert werden, um sie auch auf die übrigen Anforderungen zu testen.
Die umfangreiche Transformation (FA5) sowie der Export in eine CSV-Datei (FA6) konnte sowohl in Talend als auch in Pentaho erfolgreich umgesetzt werden. Fehlertoleranz und -behebung war in keiner Anwendung nicht vorhanden (FA7). Außerdem verlief der ETL Prozess in beiden Fällen korrekt und vollständig (NFA1), allerdings haben beide Tools ihre Vor- und Nachteile.
In Talend verlief im Gegensatz zu Pentaho der Import fehlerfrei. Außerdem benötigt Talend weniger Zeit für den Programmstart, wohingegen die Ausführung der exemplarischen Transformation in Pentaho schneller durchlief (NFA3). Die Möglichkeit in Talend eigene Funktionen mit Java Code zu schreiben ist ein weiterer Vorteil gegenüber Pentaho.
Pentaho hat zwar einen kleineren funktionalen Umfang, ist dafür aber von der Bedienung und Programmierung einfacher und intuitiver und somit wesentlich besser für Einsteiger und Menschen geeignet, die sich wenig bis gar nicht mit Programmieren auskennen. Die Anforderungen (NFA2), (NFA4) und (NFA5) werden hier also besser erfüllt.
Daher lässt sich insgesamt sagen, dass Pentaho und Talend beide sehr gut für die Datenintegration in KMU geeignet sind und es vom jeweiligen KMU, deren Mitarbeitern und insbesondere dem Anwendungsfall abhängt, welches Tool die bessere Wahl ist.

6.3.2 Fallstudie

Für die Fallstudie war insbesondere relevant, ob die Quelldatei mit Artikeldaten importiert, korrekt ins gewünschte Format transformiert und abschließend exportiert werden kann, d.h. ob (FA1) und (FA5) bis (FA7) erfüllt werden. Natürlich sollte der ETL Prozess auch korrekt und vollständig ablaufen (NFA1). Da dies wie bereits beschrieben sowohl bei Pentaho als auch bei Talend zutrifft, wären hier beide Tools gleichermaßen geeignet.
Das schnellere Programmstart-Verhalten (NFA3) von spricht im Fallbeispiel für Talend, da die Mitarbeiter so Zeit sparen würden.
Bei der Benutzer- und Einsteigerfreundlichkeit (NFA2), (NFA4) und (NFA5) hingegen kann Pentaho mehr überzeugen. Da beim Betrieb „Ihr Farbraum Metzler & Block“ möglichst auch jeder Mitarbeiter, also insbesondere Nicht-ITler, in der Lage sein sollen, den ETL Prozess auszuführen und ggf. anzupassen, ist hier Pentaho die bessere Wahl.
Insgesamt ist im Fallbeispiel also Pentaho besser geeignet, da es einerseits erfolgreich die Quelldaten ins Zielformat transformieren konnte, andererseits wesentlich benutzerfreundlicher ist und somit auch später noch von den

Mitarbeitern bedient werden kann, ohne dass diese aufwändig für das Tool geschult und eingearbeitet werden müssen.

6.3.3 Gesamtübersicht

Die folgende Tabelle fasst die Ergebnisse des Vergleichs noch einmal übersichtlich zusammen, wobei „X“ bedeutet, dass das Tool die jeweilige Anforderung erfüllt. Ein „O“ bzw. ein „-“ zeigt an, dass die Anforderung nur zum Teil bzw. gar nicht erfüllt wird.

	Pentaho	Talend
(FA1) Import XLSX	X	X
(FA2) Import CSV	X	X
(FA3) Import XML	X	X
(FA4) Import JSON	X	X
(FA5) Transformation	X	X
(FA6) Export in CSV-Datei	X	X
(FA7) Fehlertoleranz	-	-
(NFA1) Korrektheit	X	X
(NFA2) Bedienbarkeit	X	O
(NFA3) Zeitverhalten	O	X
(NFA4) Verständlichkeit	O	-
(NFA5) Erlernbarkeit	O	-

Tabelle 6.3 Gesamtübersicht zum Vergleich von Pentaho und Talend

7 Zusammenfassung

In dieser Arbeit wurde untersucht, inwieweit die beiden Tools Pentaho und Talend zur Datenintegration für KMU geeignet sind.

Dazu wurden zunächst in Kapitel 2 die Grundlagen erörtert, also der Begriff „Business Intelligence“ und semistrukturierte Daten, die später in den ETL Prozessen importiert werden sollten. Darüber hinaus wurde eine Übersicht von diversen Datenintegrationstools gezeigt sowie die später evaluierten Tools Pentaho und Talend vorgestellt. Zuletzt wurde auch Shopware beschrieben, da die später im Fallbeispiel transformierten Daten letztendlich in ein Shopware System geladen werden sollten.

Im folgenden dritten Kapitel wurden Anforderungen an Daten und Software erarbeitet, anhand derer im weiteren Verlauf der Arbeit geeignete Quelldateien erhoben und die Tools evaluiert wurden.

In Kapitel 4 wurde der Vergleich beider Tools, der Prozess der Datenerhebung sowie der umfangreiche ETL Prozess des Praxisbeispiels konzeptioniert, bei dem Artikeldaten für das KMU „Ihr Farbraum Metzler & Block“ transformiert werden sollten, sodass sie später ins Shopware Zielsystem geladen werden konnten.

Kapitel 5 behandelte die exemplarische Umsetzung des in Kapitel 4 konzeptionierten ETL Prozesses. Die Umsetzung wurde dabei sowohl in Pentaho als auch in Talend vorgestellt.

In Kapitel 6 erfolgte dann die eigentliche Bewertung. Dabei wurde zunächst beurteilt, wie gut die erhobenen Beispieldateien für die Evaluation geeignet waren,

ehe die finale Bewertung der Tools erfolgte. Außerdem wurde ein Fazit gezogen und hierbei nochmal der Vergleich zwischen Pentaho und Talend resümiert.

7.1 Ausblick

Diese Arbeit gewährt einen kleinen Einblick in den Einsatz von Tools zur Datenintegration in KMU. Dabei beleuchtet es funktionale und nicht funktionale Unterschiede zwischen Pentaho und Talend insbesondere mit Bezug auf das ausgewählte KMU. Durch die Entwicklung der ETL Prozesse in Talend und Pentaho war eine deutlich effizientere und schnellere Integration der Daten möglich. Die deutlich langsamere, fehleranfälligere und somit leistungsschwächere händische Eingabe der Daten entfiel somit.

Dennoch gibt es hier noch viele weitere Aspekte, die beleuchtet werden können. So wäre z.B. noch zu untersuchen, wie gut in den Tools das Laden aus anderen und verschiedenen Datenquellen wie Datenbanken funktioniert. Des Weiteren wäre vor allem das Zeitverhalten beider Werkzeuge bei aufwändigeren ETL Prozessen in Bezug auf größere Mengen von Datensätzen interessant zu beobachten bzw. zu vergleichen. Neben Pentaho und Talend könnten auch weitere Open Source bzw. kostengünstige Datenintegrationstools auf Ihre Eignung für KMU getestet werden.

Insgesamt bleibt die Zukunft im Bereich der Datenintegration spannend, so wird auch in KMU dieses Gebiet mehr und mehr an Bedeutung gewinnen. Das Potential der Datenintegration und -tools ist in vielen dieser Unternehmen bei weitem nicht ausgeschöpft oder zu sehr vernachlässigt. Außerdem wächst der Markt an Datenintegrationstools und diese werden dabei meist auch immer benutzerfreundlicher, sodass zukünftig möglicherweise auch Mitarbeiter von KMU ohne tiefgreifende IT-Kenntnisse in der Lage sein könnten, Daten automatisiert in bestehende oder neue Systeme zu integrieren.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Filbry u.a. 2013] FILBRY, Thomas ; GEYER, Frank ; LAUFER, Matthias ; RENKER, Sebastian ; SKOUTI, Stefan ; ROSSAK, Ines (Hrsg.): *Datenintegration : Integrationsansätze, Beispielszenarien, Problemlösungen, Talend Open Studio*. 1. Herausgabe. München : Carl Hanser Verlag, 2013

[Müller u.a. 2015] MÜLLER, Stefan ; KELLER, Christopher ; WENZKY, Sebastian (Hrsg.): *Pentaho und Jedox : Business Intelligence-Lösungen: Data Warehousing, Reporting, Analyse, Planung.* München : Carl Hanser Verlag, 2015

[Cebotarean 2011] CEBOTAREAN, Elena: Business intelligence. In: *Journal of Knowledge Management, Economics and Information Technology* (2011). Bukarest. – URL http://www.scientificpapers.org/wp-content/files/1102_Business_intelligence.pdf - Abruf: 11.07.2019

[BSH o.D.] Business Systemhaus AG (BSH AG): *Was ist Business Intelligence?* – URL https://www.bsh-ag.de/it-wissensdatenbank/business-intelligence/ - Abruf: 11.07.2019

[Vogel 2016] VOGEL Communications Group GmbH & Co. KG: *Was ist Business Intelligence? – BI?* (2016) – URL https://www.bigdata-insider.de/was-ist-business-intelligence-bi-a-563185/ - Abruf: 11.07.2019

[Gupta u.a. 2019] GUPTA, Neha ; HARE, Jim ; HUNTER, Eric ; Woodward Alys: *Market Share: Analytics and Business Intelligence, Worldwide, 2018* (2019) – URL https://www.gartner.com/en/documents/3906894 - Abruf: 12.07.2019

[Watson u.a. 2007] WATSON, Hugh ; WIXOM, Barb.: The Current State of Business Intelligence. In: *Computer* (2007), S. 96-99. – URL https://www.researchgate.net/profile/Hugh_Watson3/publication/2961945_The_Current_State_of_Business_Intelligence/links/5767e62b08aeb4b9980b0097/The-Current-State-of-Business-Intelligence.pdf - Abruf: 12.07.2019

[Chen u.a. 2012] CHEN, Hsiu-chin ; CHIANG, Roger ; STOREY, Veda.: Business Intelligence and Analytics: From Big Data to Big Impact. In: *MIS Quarterly* (2012), S. 1165-1188. – URL https://pdfs.semanticscholar.org/f5fe/b79e04b2e7b61d17a6df79a44faf358e60cd.pdf - Abruf: 12.07.2019

[Barc 2013] Business Application Research Center (BARC): *Mobile Business Intelligence – Teil 7 – Mobile BI ist die letzte Meile in der Business Intelligence Architektur* (2013) – URL https://barc.de/Artikel/mobile-business-intelligence-teil-7-mobile-bi-ist-die-letzte-meile-in-der-business-intelligence-architektur - Abruf: 13.07.2019

[Luber 2018] LUBER, Stefan: *Was ist ETL (Extract, Transform, Load)?* (2018) – URL https://www.bigdata-insider.de/was-ist-etl-extract-transform-load-a-776549/ - Abruf: 15.07.2019

[Neubert 2013] NEUBERT, Falk: *Business intelligence überblicksvortrag* (2013) – URL https://de.slideshare.net/FNeu34/business-intelligence-berblicksvortrag - Abruf: 15.07.2019

[Debitoor o.D.] DEBITOOR GmbH: *CSV-Datei - Was ist eine CSV-Datei?* – URL https://debitoor.de/lexikon/csv-datei - Abruf: 16.07.2019

[Shafranovich 2005] SHAFRANOVICH, Y.: *Common Format and MIME Type for Comma-Separated Values (CSV) Files.* Request For Comments (RFC) 4180. 2005 – URL https://tools.ietf.org/html/rfc4180 - Abruf: 16.07.2019

[Hollenbeck u.a. 2003] HOLLENBECK, S. ; ROSE, M. ; MASINTER, L.: *Guidelines for the Use of Extensible Markup Language (XML) within IETF Protocols.* Request For Comments (RFC) 3470. 2003 – URL https://tools.ietf.org/html/rfc3470 - Abruf: 17.07.2019

[Bray u.a. 2008] BRAY, Tim ; PAOLI, Jean ; SPERBERG-MCQUEEN, C.M. ; MALER, Eve ; YERGEAU, Francois: *Extensible Markup Language (XML) 1.0 (Fifth Edition).* W3C Recommendation 26 November 2008 – URL https://www.w3.org/TR/REC-xml/ - Abruf: 17.07.2019

[Bray 2017] BRAY, Tim: *The JavaScript Object Notation (JSON) Data Interchange Format.* Request For Comments (RFC) 8259. 2017 – URL https://tools.ietf.org/html/rfc8259- Abruf: 18.07.2019

[w3schools o.D.] W3SCHOOLS.com: *What is JSON?* – URL https://www.w3schools.com/whatis/whatis_json.asp - Abruf: 18.07.2019

[Augsten 2018] AUGSTEN, Stephan: *Was ist JSON?* 2018 – URL https://www.dev-insider.de/was-ist-json-a-702243/ - Abruf: 18.07.2019

[Hitachi 2019] HITACHI Vantara Corporation GmbH: *About Pentaho* – URL https://www.hitachivantara.com/go/pentaho.html - Abruf: 16.07.2019

[Luber 2017] LUBER, Stefan: *Was ist Pentaho?* (2017) – URL https://www.bigdata-insider.de/was-ist-pentaho-a-621162/ - Abruf: 16.07.2019

[Goram o.D.] GORAM, Mandy: *ETL-Werkzeug von Talend* – URL http://www.datenbanken-verstehen.de/data-warehouse/dwh-software/etl-software/talend-etl-werkzeug/ - Abruf: 17.07.2019

[Sherman 2016] SHERMAN, Rick: *Die Funktionen von Talend Enterprise Data Integration im Überblick* (2016) – URL https://www.computerweekly.com/de/ratgeber/Die-Funktionen-von-Talend-Enterprise-Data-Integration-im-Ueberblick - Abruf: 17.07.2019
[Talend 2019 I] TALEND Inc.: *Talend Homepage* (2019) – URL https://de.talend.com/ - Abruf: 17.07.2019

[Shopware o.D.] SHOPWARE AG: *Geschichte - Die perfekte Symbiose aus Technik & Design (Firmenchronik)* – URL https://www.shopware.com/de/unternehmen/story/ - Abruf: 18.07.2019

[Mittwald o.D.] MITTWALD CM Service GmbH & Co. KG: *Shopware vorgestellt* – URL https://www.mittwald.de/shopware-vorgestellt - Abruf: 18.07.2019

[Shopware 2019] SHOPWARE AG: *Import/Export: Grundmodul* – URL https://docs.shopware.com/de/shopware-5-de/import-export/import-export-grundmodul - Abruf: 18.07.2019

[Weidemann 2019] WEIDEMANN, Tobias: *Shopware: Das ändert sich mit Version 6* (2019) – URL https://t3n.de/news/shopware-aendert-version-6-1154739/ - Abruf: 18.07.2019

[Glinz 2006] GLINZ, Martin: *Requirements Engineering I – Kapitel 11: Nicht-funktionale Anforderungen* (2006) – URL https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/requirements_engineering_I_ws0607/Kapitel_11_NFAnf.pdf - Abruf: 30.07.2019

[Ahmad u.a. 2017] AHMAD, Khadija Sania ; AHMAD, Nazia ; TAHIR, Hina ; KHAN, Shaista: Fuzzy_MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements. In: *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, S. 433-437. Kannur, 2017 – URL https://ieeexplore.ieee.org/document/8342602- Abruf: 30.07.2019

[Barc 2019] Business Application Research Center (BARC): *BARC Guide Business Intelligence & Big Data 2019* (2019) – URL http://fr.zone-secure.net/52645/913200/ - Abruf: 19.07.2019

[Gartner 2018] GARTNER Inc.: *Magic Quadrant for Data Integration Tools* (2018) – URL https://www.gartner.com/en/documents/3883264/ - Abruf: 19.07.2019

[Sahu 2016] SAHU, Nisha: *Get the Row Count in PDI Dynamically* (2016) – URL https://www.helicaltech.com/row-count-in-pdi-dynamically/ - Abruf: 16.07.2019

[Pentaho 2018] PENTAHO Corporation: *Group By* (2018) – URL https://help.pentaho.com/Documentation/8.1/Products/Data_Integration/Transformation_Step_Reference/Group_By - Abruf: 16.07.2019

[Gupta u.a. 2015] GUPTA, Sarvesh ; BHATTACHARJYA, Debayan: *Data Cloning Through Pentaho Data Integration Clone Step* (2015) – URL https://www.tavant.com/blog/data-cloning-through-pentaho-data-integration-clone-step - Abruf: 16.07.2019

[Pentaho 2014] PENTAHO Community: *Add sequence* (2014) – URL https://wiki.pentaho.com/display/EAI/Add+sequence - Abruf: 17.07.2019

[Pentaho 2016] PENTAHO Corporation: *Internal Variables* (2016) – URL https://help.pentaho.com/Documentation/5.1/0L0/0Y0/090/020/010 - Abruf: 17.07.2019

[Talend o.D. I] TALEND Inc: *tFileInputExcel properties* – URL https://help.talend.com/reader/jomWd_GKqAmTZviwG_oxHQ/gLvJcNs5__whQ91iSkOgwg - Abruf: 01.08.2019

[Vogel 2019] VOGEL, Lars: *Regular expressions in Java - Tutorial* (2019), Version 3.2 – URL https://www.vogella.com/tutorials/JavaRegularExpressions/article.html - Abruf: 02.08.2019

[Talend o.D. II] TALEND Inc: *tFileOutputExcel properties* – URL https://help.talend.com/reader/jomWd_GKqAmTZviwG_oxHQ/VVY~Y6vO2ZU76p78~VHl2Q - Abruf: 02.08.2019

[Talend o.D. III] TALEND Inc: *Regular Expressions* – URL https://help.talend.com/reader/JhYq1xxY0SNSBZCbOFzZGg/W8cU~47SzE__OIkoOg5JrA - Abruf: 02.08.2019

[Talend o.D. IV] TALEND Inc: *StringHandling Routines* – URL https://help.talend.com/reader/OkqK4wnPmdoidW_TdqgMiw/o04qYO~2W55XGTDWR5y_ew - Abruf: 07.08.2019

[Talend o.D. V] TALEND Inc: *Numeric Routines* – URL https://help.talend.com/reader/1~9OqzgTIX~HofOefoxthg/THCP2Ja0ffvwXMHNIJwTTQ - Abruf: 07.08.2019

[Talend 2017] TALEND Community:
How to perform a CROSS JOIN with Talend? (2017) – URL https://community.talend.com/t5/Design-and-Development/How-to-perform-a-CROSS-JOIN-with-Talend/td-p/26164 - Abruf: 07.08.2019

[Oracle o.D.] ORACLE Corporation: *Enum RoundingMode* – URL https://docs.oracle.com/javase/8/docs/api/java/math/RoundingMode.html - Abruf: 21.08.2019

[Talend 2019 II] TALEND Community:
Create a user routine and call it in a Job (2017), Revision 4 – URL https://community.talend.com/t5/Design-and-Development/Create-a-user-routine-and-call-it-in-a-Job/ta-p/21665 - Abruf: 21.08.2019

[OpenWeather 2019] OPENWEATHER Ltd.: *OpenWeatherMap Homepage* (2019) – URL https://openweathermap.org/ - Abruf: 07.10.2019

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____ _____