



OSTBAYERISCHE  
TECHNISCHE HOCHSCHULE  
REGENSBURG

# BACHELORARBEIT

Manuel Schmidt

## Finanzberichterstattung im digitalen Zeitalter – Eine Analyse der Anwendungsmöglichkeiten des XBRL- Formates

Fakultät:  
Studiengang:  
Abgabefrist:  
Betreuerin/Prüfer:

Informatik und Mathematik  
Wirtschaftsinformatik  
19.01.2021

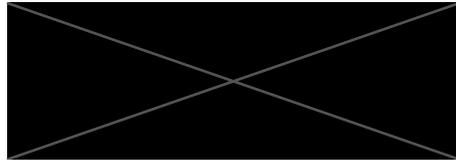


## Erklärung

1. Mir ist bekannt, dass dieses Exemplar der Bachelorarbeit/Masterarbeit als Prüfungsleistung in das Eigentum der Ostbayerischen Technischen Hochschule Regensburg übergeht.
2. Ich erkläre hiermit, dass ich diese Bachelorarbeit/Masterarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Regensburg, 13.01.2021

Ort, Datum und Unterschrift



Student: Manuel Schmidt  
Matrikelnummer:   
Bearbeitungsdauer: 19.10.2020 - 19.01.2021  
Betreuerin/Prüfer:   
Zweitprüfer: 

# Inhaltsverzeichnis

1	Kurzfassung .....	iv
2	Abbildungsverzeichnis .....	v
3	Tabellenverzeichnis .....	v
4	Code Beispiele .....	v
5	Abkürzungsverzeichnis .....	vi
6	Einleitung .....	1
6.1	Übersicht .....	2
6.2	Ziele .....	2
7	Literaturrecherche .....	2
7.1	Definition .....	2
7.2	Problemstellung .....	3
7.3	Entwicklung des XBRL Standards .....	5
7.4	Nutzergruppen von XBRL .....	6
7.4.1	Unternehmen (Producer) .....	7
7.4.2	Banken & Kreditgeber (Consumer) .....	7
7.4.3	Statistikämter & Regulatoren (Consumer) .....	7
7.4.4	Investoren (Consumer) .....	8
7.5	Der grobe Aufbau von XBRL .....	8
7.6	XML-Schicht .....	9
7.6.1	Namespaces .....	9
7.6.2	XLink .....	10
7.7	XBRL-Spezifikation und XBRL-Taxonomien .....	12
7.7.1	Taxonomie-Schema .....	13
7.7.2	Taxonomie-Linkbases .....	14
7.7.3	Erweiterbarkeit von Taxonomien .....	20
7.7.4	Modularisierung von Taxonomien .....	22
7.8	Instance Documents .....	22
7.8.1	XBRL Instance Documents .....	23
7.8.2	Unterschiede bei iXBRL .....	25
7.9	Übersicht über einen vollständigen Bericht .....	26
8	Anwendungsbeispiel .....	28
8.1	Ziele des Projektes .....	28
8.2	Open Source Lösungen .....	28
8.3	Aufbau des Projektes .....	29

---

8.4	SEC EDGAR.....	30
8.4.1	Aufbau der Jahres- und Quartalsberichte.....	31
8.4.2	Schnittstellen von EDGAR.....	31
8.5	Anwendungsschicht.....	32
8.5.1	Einlesen des Structured Disclosure RSS Feeds von EDGAR.....	32
8.5.2	Caching Strategie.....	34
8.5.3	Aufbau und Funktionsweise des Parsers.....	34
8.5.4	Zwischenspeichern der Facts.....	40
8.5.5	Beispielhafte Analyse.....	41
9	Kritische Betrachtung.....	44
10	Fazit.....	45
11	Literaturverzeichnis.....	46
12	Anhang.....	49

# 1 Kurzfassung

Mit der Einführung des „European Single Electronic Format“ (ESEF) am 01. Januar 2020 sind nun auch kapitalmarktorientierte Unternehmen innerhalb der EU dazu verpflichtet, ihre Jahresabschlüsse mit dem technischen Standard XBRL (eXtensible Business Reporting Language) einreichen. Ziel dieser neuen Vorgaben ist die Verbesserung der Analysierbarkeit und Vergleichbarkeit von Jahresfinanzberichten, zugunsten von Investoren, Behörden und der Unternehmen selbst.<sup>1</sup>

Diese Bachelorarbeit soll die unterliegenden technischen Spezifikationen des XBRL Standards beleuchten und deren Anwendungsmöglichkeiten und Vorteile für die verschiedenen Stakeholder aufzeigen. Anhand eines praxisorientierten Beispiels soll gezeigt werden, wie XBRL Berichte maschinell eingelesen und analysiert werden können. Dabei sollen eventuelle Hürden bei der maschinellen Verarbeitung erarbeitet werden.

---

<sup>1</sup> Vgl. European Securities and Markets Authority: Final Report on the RTS on the European Single Electronic Format, in: ESMA Website, 18.12.2017, [online]  
[https://www.esma.europa.eu/sites/default/files/library/esma32-60-204\\_final\\_report\\_on\\_rts\\_on\\_esef.pdf](https://www.esma.europa.eu/sites/default/files/library/esma32-60-204_final_report_on_rts_on_esef.pdf) [26.11.2020], S.3.

## 2 Abbildungsverzeichnis

Abbildung 1: Wichtige Informationen im Zusammenhang mit einem Finanzbericht .....	4
Abbildung 2: Die Schichten von XBRL.....	9
Abbildung 3: Strukturelle Darstellung einer Referenz Linkbase .....	15
Abbildung 4: Ausschnitt aus §266 Absatz 2 HGB .....	17
Abbildung 5: Strukturelle Darstellung einer Label Linkbase .....	18
Abbildung 6: Gegenüberstellung der offenen und der geschlossenen Berichtserstattung...21	
Abbildung 7: Struktur eines Instance Documents mit Taxonomie .....	27
Abbildung 8: Komponentendiagramm des Projektes .....	30
Abbildung 9: Ausschnitt einer Index Datei des EDGAR Systems.....	31
Abbildung 10: Struktur der RSS Index Tabellen.....	33
Abbildung 11: Aktivitätsdiagramm des XBRL-Parsers .....	35
Abbildung 12: Ablauf beim Einlesen der Taxonomie-Importe.....	36
Abbildung 13: Hierarchische Struktur einer eingelesenen Calculation Linkbase .....	38
Abbildung 14: Struktur der Tabellen zum Zwischenspeichern.....	40
Abbildung 15: Formel des KGV .....	41
Abbildung 16: Ergebnis der KGV-Analyse .....	43
Abbildung 17: Klassendiagramm des XBRL-Parsers.....	49

## 3 Tabellenverzeichnis

Tabelle 1: Übersicht über die wichtigsten Label Role Typen.....	19
Tabelle 2: Übersicht über die wichtigsten Reference Role Typen .....	20
Tabelle 3: Erläuterung der Spalten der sub_index_items Tabelle .....	33

## 4 Code Beispiele

Codebeispiel 1: Setzen von XML Namespaces .....	10
Codebeispiel 2: Beispiel eines erweiterten XLinks.....	11
Codebeispiel 3: Deklaration eines Konzeptes am Beispiel der IFRS Taxonomie .....	13
Codebeispiel 4: Einbettung einer Linkbase in ein Taxonomie-Schema.....	14
Codebeispiel 5: Verlinkung von Konzepten in der Calculation Linkbase .....	16
Codebeispiel 6: Arcs in der Presentation Linkbase.....	17
Codebeispiel 7: Verlinkung von Bezeichnungen in der Label Linkbase .....	19
Codebeispiel 8: Verlinkung von Literatur in der Referenz Linkbase .....	20
Codebeispiel 9: Einbindung der Taxonomie in das Instance Document.....	23
Codebeispiel 10: Erstellung eines Facts in einem Instance Document .....	23
Codebeispiel 11: Variationen bei der Erstellung von XBRL-Einheiten.....	23
Codebeispiel 12: Variation bei der Erstellung von Kontext.....	24
Codebeispiel 13: Unterschiede bei der Fact-Deklaration zwischen XBRL und iXBRL.....	25
Codebeispiel 14: Auszug des monatlichen RSS Feeds von EDGAR.....	32
Codebeispiel 15: Einlesen der importierten Taxonomien .....	37
Codebeispiel 16: Deklaration eines Facts.....	39
Codebeispiel 17: Funktion zur rekursiven Suche einer Taxonomie.....	39
Codebeispiel 18: Abrufen der für den Fact benötigten Informationen .....	39
Codebeispiel 19: SQL-Befehl zum Abrufen von Gewinn pro Aktien und Aktienkurs.....	42

## 5 Abkürzungsverzeichnis

10-K	-	Bezeichnung der SEC für einen Jahresbericht
10-Q	-	Bezeichnung der SEC für einen Quartalsbericht
AICPA	-	American Institute of Certified Public Accountants
API	-	Application Programming Interface
CIK	-	Central Index Key
COREP	-	Common Reporting
CPA	-	Certified Public Accountant
DTD	-	Document Type Definition
EDGAR	-	Electronic Data Gathering, Analysis, and Retrieval system
ERP	-	Enterprise resource planning
ESEF	-	European Single Electronic Format
ESMA	-	European Securities and Markets Authority
FYE	-	Fiscal Year End (Ende des Geschäftsjahres)
GAAP	-	Generally Accepted Accounting Principles
GCD	-	Global Common Data/Document
GuV	-	Gewinn- und Verlustrechnung
HGB	-	Handelsgesetzbuch
HTML	-	Hypertext Markup Language
IFRS	-	International Financial Reporting Standards
iXBRL	-	inline eXtensible Business Reporting Language
JSON	-	JavaScript Object Notation
KGV	-	Kurs Gewinn Verhältnis
LEI	-	Legal Entity Identifier
PyPI	-	Python Package Index
RDF	-	Resource Description Framework
RSS	-	RDF Site Summary
RTS	-	Regulatory Technical Standards
SEC	-	Securities and Exchange Commission
SIC	-	Standard Industrial Classification
W3C	-	World Wide Web Consortium
XBRL	-	eXtensible Business Reporting Language
XBRL GL	-	XBRL Global Ledger
XHTML	-	Extensible Hypertext Markup Language
XLink	-	XML Linking Language
XML	-	Extensible Markup Language
XMLNS	-	XML Namespace
XPath	-	XML Path Language
XPointer	-	XML Pointer Language

## 6 Einleitung

Seit einigen Jahren sind wir dank der rasant voranschreitenden Digitalisierung den automatisierten Datenaustausch zwischen zwei Parteien in vielen computergestützten Anwendungsbereichen gewohnt. So ist es beispielsweise eine Selbstverständlichkeit geworden, dass ein Onlinehandel nach Eingang einer Bestellung automatisch den Zahlungsdienstleister kontaktiert und den Zahlungsvorgang durchführt. Der Kunde muss bis auf das Bereitstellen seiner Kontoinformationen und seines Einverständnisses keine weiteren Aktionen durchführen, um die Zahlung erfolgreich abzuschließen.

Dieser Vorgang ist nur aufgrund eines standardisierten Informationsaustausches zwischen dem Onlinehandel bzw. dem Zahlungsabwickler und der betroffenen Bank möglich und wird heutzutage von dem Kunden als selbstverständlich angesehen. Bei dem Informationsaustausch zwischen Unternehmen bzw. zwischen Unternehmen und Aufsichtsbehörden ist dieser reibungslose, zwischenparteiische Informationsaustausch aber oft keine Selbstverständlichkeit. Hier findet man häufig noch Prozesse, die aufgrund von vielen manuellen Prozessschritten ineffizient und oft auch intransparent ablaufen. Ein Beispiel hierfür ist die finanzielle Berichterstattung von publikationspflichtigen Unternehmen.

Innerhalb des unternehmensinternen Rechnungswesens werden Buchhaltungssysteme verwendet, welche die Bewegungen in einem Unternehmen strukturiert erfassen und abspeichern. Der darauf aufbauende Jahresabschluss wurde in der Vergangenheit aber meist entweder im PDF-Format oder entsprechender Papierform veröffentlicht, wodurch die Datenpunkte innerhalb des Geschäftsberichtes jegliche Systematik und Struktur verlieren.

Dies zieht vor allem für die Rezipienten des Geschäftsberichtes erhebliche Nachteile mit sich. Um die Daten aus dem Jahresabschluss maschinell verarbeiten und vergleichen zu können, müssen diese erst wieder aufwendig von dem PDF-Format in eine strukturierte Datenumgebung überführt werden.

Dieses Problem versucht XBRL (**eXtensible Businesses Reporting Language**) aufzugreifen und zu lösen. Der Standard wird schon von über 100 Regulierungsbehörden in mehr als 60 Ländern verwendet.<sup>2</sup> Seit dem 1. Januar 2020 hat nun auch die Europäische Union eine EU-weite Richtlinie zur Implementierung des XBRL-Standards in der Finanzberichterstattung von europäischen, kapitalmarktorientierten Unternehmen in Kraft gesetzt.<sup>3</sup> Diese wurde anschließend im Juni 2020 in entsprechendes deutsches Recht umgesetzt (§328 HGB).

---

<sup>2</sup> Vgl. Nitchman, Dave: XBRL Around the World, in: XBRL International Inc, 30.09.2016, [online] <https://www.xbrl.org/xbrl-around-the-world> [25.11.2020].

<sup>3</sup> Vgl. European Securities and Markets Authority, 2017, S.3

## 6.1 Übersicht

Die vorliegende Bachelorarbeit ist in drei große Teile aufgebaut. Im ersten Teil wird auf den technischen Aufbau des XBRL-Standards eingegangen und erklärt, aus welchen Elementen ein mit XBRL erstellter Finanzbericht besteht. Im zweiten Teil wird anhand eines technischen Anwendungsbeispiels gezeigt, wie Daten aus XBRL-Berichten ausgelesen und weiterverarbeitet werden können. Abschließend geht der dritte Teil auf die Ergebnisse dieser Bachelorarbeit ein und schießt die Arbeit mit einem Fazit ab.

## 6.2 Ziele

Ziel dieser Arbeit ist, den technischen Aufbau des XBRL Standards zu erklären und durch die Implementierung eines XBRL-Parsers zu evaluieren, wie auf XBRL basierende Finanzberichte maschinell ausgelesen werden können. Dabei soll der Fokus primär auf der Frage liegen, wie XBRL Berichte für spätere unternehmensübergreifende Analysen ausgewertet werden können und welche Hürden während des Analysierprozesses auftreten.

# 7 Literaturrecherche

## 7.1 Definition

Wie bereits in der Einleitung beschrieben ist XBRL ein globaler Standard im Berichts- und Meldewesen. Er wurde speziell für die Übertragung von Finanzinformationen entwickelt und optimiert.

In dem Buch „New Dimensions of Business Reporting and XBRL“ beschreiben Debreceny et al. den Standard als:

„eXtensible Businesses Reporting Language (XBRL) is established as a standard that supports intra- and inter-enterprise reporting as well as to a variety of information consumers. A key objective of the XBRL standard is to increase the efficiency of the usage of information systems at the interface of business management and information technology.“<sup>4</sup>

Eine technischere Definition gibt Althoff in „Die neue E-Bilanz“:

„XBRL (eXtensible Business Reporting Language) basiert auf der eXtensible Markup Language (XML) und wurde speziell zum Datenaustausch von Finanzdaten entwickelt. Bestehende XML-Spezifikationen wurden dabei explizit für diesen Bereich erweitert.

---

<sup>4</sup> Debreceny, Roger/Carsten Felden/Bartosz Ochocki/Maciej Piechocki/Michal Piechocki: XBRL for Interactive Data: Engineering the Information Value Chain, 2009. Aufl., Heidelberg, Deutschland: Springer, Berlin, Heidelberg, 2009, S. V.

XBRL selbst stellt lediglich das Regel- oder Grundgerüst (Sprach- oder Syntaxregeln) dar, nach dem dann eine Datenübermittlung erfolgen kann.“<sup>5</sup>

Zusammenfassend lässt sich XBRL als eine etablierte, auf XML basierende Spezifikation, welche zur Übertragung von Finanzdaten genutzt wird, definieren.

Da es eine Vielzahl von Finanzinformationen mit unterschiedlichen Rechnungslegungssystemen gibt, muss vor der Datenübertragung ein gemeinsames Sprachgerüst aufgesetzt werden. Dieses gemeinsame Sprachgerüst wird im XBRL-Kontext **Taxonomy (Taxonomie)** genannt. Eine Taxonomie beschreibt, welche Daten übertragen werden können und gibt weiterführende Informationen zu den einzelnen Datenpunkten.<sup>6</sup>

Der kleinste Teil einer Taxonomie sind die **Concepts (Konzepte)**. Ein Konzept beschreibt eine einzelne fachspezifische Dateneinheit, mit welcher ein Wert im späteren Finanzbericht gekennzeichnet werden kann. Die HGB-Taxonomie, welche für die Übertragung von Jahresabschlüssen von Unternehmen entwickelt worden ist, enthält z.B. die Konzepte „Umlaufvermögen“, „Verbindlichkeiten aus Lieferung & Leistungen“, „Gewinnrücklagen“ und andere für den Jahresabschluss relevante Positionen.<sup>7</sup>

Der zweite große Baustein neben der Taxonomie sind die **Instance Documents**. Während die Taxonomien meistens von Regulatoren oder Standardisierungsgremien herausgegeben werden, werden die *Instance Documents* von dem berichtenden Unternehmen verfasst. Sie enthalten die eigentlichen Daten, die zu übertragen sind und werden mit den Konzepten aus einer unterliegenden Taxonomie gekennzeichnet. Der genauere technische Aufbau wird in den späteren Kapiteln weiter besprochen.

## 7.2 Problemstellung

Eingangs wurde bereits erwähnt, dass einer der zentralen Ziele von XBRL die standardisierte und strukturierte Übermittlung von Finanzinformationen ist. Aber warum ist dies überhaupt eine Herausforderung?

Finanzberichte enthalten sehr viele Metainformationen wie Abbildung 1 zeigt. In dieser Kapitalflussrechnung der Volkswagen AG wurden wichtige Metainformationen hervorgehoben. Diese lassen sich für den menschlichen Betrachter zwar durch Aufbau und Kontext erschließen, sind aber aufgrund der unterschiedlichen Layouts und Strukturen von Finanzberichten sehr schwer maschinell auszulesen. Wenn uneinheitliche Berichte auch noch von unterschiedlichen Unternehmen verfasst werden, wird das Auslesen der Daten besonders schwer. Gerade im Meldewesen, wo Behörden teilweise Finanzberichte von allen Unternehmen eines Landes verarbeiten müssen, kann dies zum Problem werden.

---

<sup>5</sup> Vgl. Althoff, Frank/Andreas Arnold/Arne Jansen/Tobias Polka/Frank Wetzel: Die neue E-Bilanz, 1. Aufl., Freiburg, Deutschland: Haufe-Lexware GmbH, 2011, S. 53.

<sup>6</sup> Vgl. Althoff et al., 2011, S. 53.

<sup>7</sup> Vgl. Althoff et al., 2011, S. 53.

Um einen Datenpunkt aus der Kapitalflussrechnung (Abb. 1) herauslesen und interpretieren zu können, ist ein Blick auf verschiedene Orte des Berichtes erforderlich. So benötigt man beispielsweise für den rot markierten Wert „2200“ Informationen über die Größenordnung (hier Millionen), die Währung (€), den zeitlichen Bezug (01.01.2020 – 30.09.2020) und die buchhalterische Position („Change in liabilities“). Außerdem können sich Werte in der Kapitalflussrechnung nur auf einen Bereich des Unternehmens beziehen, wie hier der Bereich „Automotive“.

		VOLKSWAGEN GROUP		AUTOMOTIVE <sup>1</sup>		FINANCIAL SERVICES	
		2020	2019	2020	2019	2020	2019
€ million							
Cash flow statement for the Period January 1 to September 30							
Welche Währung?							
Welcher Finanzbericht?							
Welche Zeitperiode?							
Welche Größenordnung?							
Welche Firma?							
Welche(r/s) Bereich/ Subunternehmen?							
Welches Jahr?							
Welches Konzept?							
Welcher mathematischer Zusammenhang?							
Nach welchen Standards wird die Zahl berechnet? (IFRS/ GAAP...)							
Irgendwelche Anmerkungen?							
Cash and cash equivalents at beginning of period		24,329	28,113	18,098	23,354	6,231	4,759
Earnings before tax		2,254	14,637	441	12,378	1,813	2,259
Income taxes paid		-1,862	-2,112	-1,408	-1,561	-454	-551
Depreciation and amortization expense <sup>2</sup>		19,747	18,297	12,779	12,016	6,968	6,281
Change in pension provisions		649	280	636	265	13	15
Share of the result of equity-accounted investees		828	-249	872	-197	-43	-52
Other noncash income/expense and reclassifications		-505	-955	-313	-577	-193	-377
<b>Gross cash flow</b>		<b>21,111</b>	<b>29,898</b>	<b>13,007</b>	<b>22,323</b>	<b>8,104</b>	<b>7,575</b>
Change in working capital		-2,812	-17,139	164	-1,402	-2,977	-15,737
Change in inventories		1,570	-3,725	1,134	-3,699	435	-27
Change in receivables		-2,032	-2,495	-1,784	-2,602	-248	108
Change in liabilities		3,470	5,729	2,200	4,925	1,270	805
Change in other provisions		-1,300	-11	-1,409	51	109	-62
Change in lease assets (excluding depreciation)		-8,770	-9,552	187	-80	-8,957	-9,472
Change in financial services receivables		4,249	-7,086	-164	4	4,414	-7,089
<b>Cash flows from operating activities</b>		<b>18,298</b>	<b>12,759</b>	<b>13,171</b>	<b>20,921</b>	<b>5,127</b>	<b>-8,162</b>
Cash flows from investing activities attributable to operating activities		-11,784	-12,532	-11,754	-12,309	-30	-223
of which: Investments in intangible assets (excluding capitalized development)		-6,551	-8,344	-6,422	-8,186	-129	-158
Capitalized development		-4,481	-3,667	-4,481	-3,667	-	-
Acquisition and disposal of equity investments		-1,024	-677	-1,096	-595	72	-82
<b>Net cash flow<sup>4</sup></b>		<b>6,514</b>	<b>227</b>	<b>1,418</b>	<b>8,612</b>	<b>5,097</b>	<b>-8,385</b>
Change in investments in securities, loans and time deposits		-2,913	1,007	-2,890	-3,510	-22	4,516
<b>Cash flows from investing activities</b>		<b>-14,697</b>	<b>-11,525</b>	<b>-14,644</b>	<b>-15,818</b>	<b>-53</b>	<b>4,293</b>
Cash flows from financing activities		17,835	748	14,178	-6,985	3,657	7,733
of which: capital transactions with noncontrolling interests		-	1,368	-	1,368	-	-
capital contributions/capital redemptions		2,984	-	-	-	-	981
MAN noncontrolling interest shareholders: compensation payments and acquisition of shares tendered		2	-	-	-	-	-
Effect of exchange rate changes on cash and cash equivalents		-658	-	-	-	-	59
Change of loss allowance within cash & cash equivalents		-2	0	-2	0	0	0
<b>Net change in cash and cash equivalents</b>		<b>20,777</b>	<b>2,284</b>	<b>12,158</b>	<b>-1,638</b>	<b>8,619</b>	<b>3,923</b>
Cash and cash equivalents at Sept. 30 <sup>5</sup>		45,107	30,397	30,256	21,715	14,850	8,682
Securities, loans and time deposits		31,217	27,375	15,901	12,319	15,316	15,056
<b>Gross liquidity</b>		<b>76,324</b>	<b>57,772</b>	<b>46,157</b>	<b>34,034</b>	<b>30,166</b>	<b>23,738</b>
Total third-party borrowings		-210,801	-202,515	-21,310	-14,247	-189,491	-188,268
<b>Net liquidity at Sept. 30<sup>6</sup></b>		<b>-134,477</b>	<b>-144,743</b>	<b>24,848</b>	<b>19,787</b>	<b>-159,325</b>	<b>-164,530</b>
For information purposes: at Jan. 1		-148,040	-134,735	21,276	19,368	-169,316	-154,103

Abbildung 1: Wichtige Informationen im Zusammenhang mit einem Finanzbericht<sup>8 9</sup>

<sup>8</sup> Eigene Darstellung nach Debreceny et al., 2009, S. 5.

<sup>9</sup> Basierend auf Volkswagen AG: Interim Report Q3 2020, in: Volkswagen AG, 29.10.2020, [online] [https://www.volkswagenag.com/presence/investorrelation/publications/interim-reports/2020/Q3\\_2020\\_e.pdf](https://www.volkswagenag.com/presence/investorrelation/publications/interim-reports/2020/Q3_2020_e.pdf) [01.12.2020] S.36.

### 7.3 Entwicklung des XBRL Standards

Bevor die technischen Aspekte beleuchtet werden und der aktuelle Stand der Technologie vorgestellt wird, sollte ein Blick auf die Vergangenheit geworfen werden, um die Entwicklung besser verstehen und einordnen zu können.

Als Charles Hoffman, der oft auch als „Vater von XBRL“ bezeichnet wird, im April 1998 das erste Mal von XML gehört hatte, kam ihm die Idee, diese Technologie auch im externen Rechnungswesen anzuwenden. Er selbst arbeite schon seit über 15 Jahren als Certified Public Accountant (CPA – lizenziertes Wirtschaftsprüfer) im externen Rechnungswesen und beschäftigte sich ein Großteil seiner Zeit mit der Integration und Kommunikation zweier Informationssysteme.<sup>10</sup>

Seine Idee wurde von Anfang an vom “American Institute of Certified Public Accountants” (AICPA) unterstützt. Nach der Erstellung eines ersten Prototyps wurde ein Lenkungsausschuss gegründet, welcher die Aufgabe der Weiterentwicklung von XBRL besaß. Der Lenkungsausschuss umfasste 12 Mitglieder im Jahr 1998 und wuchs bis zum Jahr 2003 auf 170 Mitglieder an. In ihm befanden sich auch viele führende Buchhaltungs-, Technologie- und Regierungsorganisationen.<sup>11</sup> Des Weiteren bildeten sich lokale Konsortien sog. „Jurisdictions“, welche von XBRL International als offizielle Vertreter ihres Landes im internationalen Konsortium anerkannt werden.

Am 16. März 2005 ermöglichte die **Securities and Exchange Commission (SEC)** aus den Vereinigten Staaten Unternehmen zusätzlich mit XBRL markierte Finanzinformationen auf freiwilliger Basis einzureichen.<sup>12</sup> Im April 2009 wurde das Einreichen von Jahres- und Quartalsabschlüssen im XBRL-Format verpflichtend gemacht. Von dort an mussten in den USA alle Firmen und Investmentfonds mit Vermögenswerten über \$10 Mio. ihre Finanzberichte zusätzlich im XBRL-Format abgeben.<sup>13</sup>

Parallel dazu liefen in der europäischen Union auch einige XBRL Projekte. Bereits im Jahr 2008 hat das Europäische Parlament die Bedeutung von XBRL für ein transparentes und effizientes Berichtswesen betont und die Europäische Kommission aufgefordert, einen Fahrplan für die Einführung der XBRL-Berichterstattung von börsennotierten Unternehmen vorzulegen.<sup>14</sup> Daraufhin wurde die Technologie von verschiedensten Organisationen in der EU adoptiert.

<sup>10</sup> Vgl. Debreceny et al., 2009, S. XXVII.

<sup>11</sup> Vgl. Wu, Jia/Miklos Vasarhelyi: XBRL: A New Tool For Electronic Financial Reporting, in: Business Intelligence Techniques, 2004, doi: 10.1007/978-3-540-24700-5\_5, S. 78.

<sup>12</sup> Vgl. U.S. Securities and Exchange Commission: FINAL RULE: XBRL Voluntary Financial Reporting Program on the EDGAR System, in: SEC Website, 02.03.2005, [online] <https://www.sec.gov/rules/final/33-8529.htm> [26.11.2020].

<sup>13</sup> Vgl. U.S. Securities and Exchange Commission: Interactive Data to Improve Financial Reporting, in: SEC Website, 30.01.2009, [online] <https://www.sec.gov/rules/final/2009/33-9002.pdf> [26.11.2020].

<sup>14</sup> Vgl. Europäisches Parlament: A simplified business environment for companies - P6\_TA(2008)0220, in: Website des Europäischen Parlamentes, 21.05.2008, [online] <https://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+TA+P6-TA-2008-0220+0+DOC+XML+V0//EN> [01.12.2020].

So hat beispielsweise die Deutsche Bundesbank die Berichterstattung über Verbriefungen von Krediten und Forderungen auf XBRL umgestellt.<sup>15</sup> Ein weiteres Beispiel ist die Umstellung auf den XBRL-Standard bei eingehenden Finanzberichten unterschiedlicher Firmen und Institutionen bei der spanischen Zentralbank.<sup>16</sup>

In Deutschland startete das Bundesfinanzministerium im Jahr 2010 mit dem Projekt „**E-Bilanz**“ eine Initiative, bei der verschiedene Verfahren zur elektronischen Übermittlung von Jahresabschlussunterlagen an die Finanzverwaltung evaluiert werden sollten.<sup>17</sup> Auch hier wurde XBRL als zukünftiger Standard festgelegt und ab 2013 verpflichtend für alle bilanzierenden Unternehmen eingeführt.<sup>18</sup> Seitdem sind die Handelsbilanz/Steuerbilanz, die GuV-Rechnung und die Ergebnisverwendungsrechnung in diesem Format dem Finanzamt einzureichen (§5b EStG). Die eingereichten Dokumente dienen allerdings ausschließlich dem Besteuerungsverfahren und unterliegen somit dem Steuergeheimnis.

Mit dem Regulierungsstandard „**European Single Electronic Format**“ (ESEF) der europäischen Wertpapier- und Marktaufsichtsbehörde (ESMA) gibt es nun auch eine EU-weite Richtlinie für die Berichterstattung im XBRL-Standard. Diese gilt für alle Unternehmen die Wertpapiere innerhalb der EU emittiert haben und wurde im Juni 2020 in deutsches Recht umgesetzt (§328 HGB). Die Entwicklung des ESEF wurde vom Europäischen Parlament gefordert. Dadurch wird eine Verbesserung der Zugänglichkeit, der Analyse und der Vergleichbarkeit von Jahresberichten für Emittenten, Anleger und Behörden erhofft (Richtlinie 2013/50/EU Artikel 26). Im Gegensatz zur E-Bilanz, welche ausschließlich dem Besteuerungsverfahren dient, ist das ESEF also eher kapitalmarktorientiert.

## 7.4 Nutzergruppen von XBRL

Die Nutzergruppen eines XBRL-basierten Informationsflusses kann man in zwei Gruppen aufteilen. Die „**Producer**“ sind alle Personen, die bei der Erstellung eines Finanzberichtes mitwirken. Die „**Consumer**“ sind alle Personen, welche die Finanzberichte entgegennehmen, weiterverarbeiten oder auswerten.<sup>19</sup> Wichtig zu beachten ist, dass die „Consumer“ sowohl intern als auch extern sein können. Wenn ein Unternehmen mehrere Niederlassungen oder Subunternehmen mit eigenen IT-Systemen umfasst, können wichtige Informationen mithilfe von XBRL an eine zentrale Abteilung übermittelt und dort gebündelt werden.<sup>20</sup> Der „Consumer“ wäre dann die unternehmensinterne Abteilung, welche die Informationen zusammenträgt und beispielsweise im Rahmen der Berichtspflicht an die Behörden weitergibt oder anderweitig auswertet. Im Folgenden soll auf ein paar selektierte Producer und Consumer genauer eingegangen werden.

<sup>15</sup> Vgl. Enachi, Mihaela/Ioan I. Andone: The Progress of XBRL in Europe – Projects, Users and Prospects, in: Procedia Economics and Finance, Jg. 2015, Nr. 20, 2015, doi: 10.1016/s2212-5671(15)00064-7, S. 188.

<sup>16</sup> Vgl. Enachi/Andone, 2015, S. 189.

<sup>17</sup> Vgl. Flickinger, 2013, S. 12.

<sup>18</sup> Vgl. Flickinger, 2013, S. 118-119.

<sup>19</sup> Vgl. Flickinger, 2013, S. 18.

<sup>20</sup> Vgl. Flickinger, 2013, S. 22.

### 7.4.1 Unternehmen (Producer)

Unternehmen müssen verschiedene Berichte an unterschiedliche Entitäten herausgeben. So müssen Kreditberichte für die Beantragung von neuen Bankkrediten erstellt, Jahresberichte für Investoren auf der Firmenwebseite veröffentlicht, sowie die nach §325 HGB vorgeschriebenen Unterlagen an die Behörden übermittelt werden. All diese Berichte haben unterschiedliche Formatierungsanforderungen, benötigen aber ähnliche Informationen über das Unternehmen.

Das manuelle Erstellen dieser Berichte ist nicht nur aufwendig, sondern ist auch oft mit Übertragungsfehlern und Unstimmigkeiten verbunden. Wenn die Finanzdaten aber bereits im XBRL-Format vorliegen, können sie in ein System eingelesen und automatisch in die verschiedenen Berichte eingefügt werden. Dadurch könnte der Berichtserstellungsprozess der Buchhaltung effizienter gemacht werden.<sup>21</sup>

Mit der Einführung von XBRL Global Ledger (XBRL GL) wurde zudem ein weiterer Standard entwickelt, mit dem auch die transaktionsorientierte Architektur eines ERP-Systems abgebildet werden kann.<sup>22</sup> Mit einer solchen Schnittstelle wäre auch eine vollständige Automatisierung des Berichts-Erstellungsprozesses möglich. Da XBRL GL unabhängig von ERP-Systemen ist, ist auch ein Austausch von Daten unterschiedlicher ERP-Systeme möglich.<sup>23</sup> XBRL GL ist allerdings kein Bestandteil der klassischen XBRL-Spezifikation (XBRL 2.1), deshalb wird in dieser Arbeit nicht weiter darauf eingegangen.

### 7.4.2 Banken & Kreditgeber (Consumer)

Der Bankensektor ist in Deutschland einer der führenden Adressatenkreise, wenn es um die Adaption von XBRL geht. Schon im Jahre 2009 war es für die Kreditkunden der Deutschen Bank möglich, Finanzdaten im XBRL Format einzureichen und somit den Kreditvergabeprozess zu beschleunigen.<sup>24</sup> Erreicht wird die Beschleunigung durch die Benutzung des XBRL Standards bei der Bonitätsprüfung. Dadurch kann eine automatisierte Analyse der Finanzdaten vollzogen werden, womit die Effizienz des Prozesses stark erhöht wird.

### 7.4.3 Statistikämter & Regulatoren (Consumer)

Statistikämter wie das Statistische Bundesamt in Deutschland sind für die Erhebung und Veröffentlichung von wichtigen nationalen Statistiken verantwortlich. Darunter fallen auch ausschlaggebende Wirtschaftsindikatoren wie beispielsweise das Bruttoinlandsprodukt, die volkswirtschaftliche Gesamtrechnung, sowie einzelne Statistiken und Indikatoren zu verschiedenen Sektoren innerhalb des eigenen Landes.

---

<sup>21</sup> Vgl. Wu/Vasarhelyi, 2004, S. 85.

<sup>22</sup> Vgl. Debreceny et al., 2009, S. 71.

<sup>23</sup> Vgl. Flickinger, 2013, S. 50.

<sup>24</sup> Vgl. Pfister, Jan/Sandro Venetz: Die Anwendung von XBRL in IR, in: Klaus Rainer Kirchhoff/Manfred Piwinger (Hrsg.), Praxishandbuch Investor Relations, 2., überarb. u. erw. Aufl. 2009., Wiesbaden, Deutschland: Gabler Verlag, 2009, S. 437.

Um diese Statistiken aufstellen zu können, müssen Daten über die wirtschaftlichen Aktivitäten von den verschiedenen Unternehmen des Landes gesammelt und verarbeitet werden.<sup>25</sup>

Aktuell werden diese Daten oft mit elektronischen oder papierbasierten Formularen von den Firmen an die Statistikämter übertragen.<sup>26</sup> Allerdings ergreifen Regulatoren und Statistikämter Maßnahmen, um diesen Prozess effizienter abzudecken.

So startete die Europäische Kommission im Jahr 2012 ein Pilotprojekt, mit welchem die Entwicklung von IT-Systemen für die effizientere Erfassung statistischer Daten gefördert werden sollte. Im Rahmen dieses Projektes wurden XBRL-Taxonomien erstellt, welche speziell für die Erfassung der von den Statistikämtern benötigten Daten optimiert sind.<sup>27</sup>

#### 7.4.4 Investoren (Consumer)

Um eine gute Investmententscheidung treffen zu können benötigt es einer vorausgehenden Analyse der infrage stehenden Werte. Hierfür braucht der Investor qualitative und vergleichbare Finanzdaten. XBRL kann den Zugang zu diesen Finanzdaten erleichtern, da Berichte im XBRL-Format aufgrund des einheitlichen Standards einfacher zu vergleichen sind.<sup>28</sup> „Hinsichtlich Glaubwürdigkeit, Wesentlichkeit und Transparenz der Finanzinformation würde XBRL [...] einen wichtigen Beitrag liefern“.<sup>29</sup>

### 7.5 Der grobe Aufbau von XBRL

Die Begriffe Taxonomie und *Instance Document* wurden in der Einleitung schon kurz erläutert. Abbildung 2 zeigt den Zusammenhang dieser Begriffe auf. Das *Instance Document* in der obersten Schicht wird von dem betroffenen Unternehmen verfasst und enthält die eigentlichen Daten, die das Unternehmen veröffentlichen will. Es baut auf der Taxonomie auf, welche meist von den Aufsichtsbehörden selbst oder von Standardisierungsgremien erstellt wird. Die Taxonomie listet alle Konzepte auf, die in dem *Instance Document* verwendet werden dürfen.

Eine Schicht darunter liegt die **XBRL-Spezifikation**. Sie beschreibt den Aufbau und die Erstellung des darüberliegenden *Instance Documents* und der Taxonomie. Zudem optimiert die XBRL-Spezifikation das klassische XML für die Übermittlung von Finanzinformationen, indem sie darunterliegende XML-Elemente modifiziert und weitere fachspezifische Elemente hinzufügt.

---

<sup>25</sup> Vgl. Roos, Marko: The Dutch Taxonomy Project and structural regulatory business reporting: impact for Statistics Netherlands (Draft), in: Website der Europäischen Kommission, [online] <https://ec.europa.eu/eurostat/documents/1001617/4411693/II-1-NL-DUTCH-TAXONOMY-PROJECT.pdf> [10.12.2020], S.1.

<sup>26</sup> Vgl. Roos, o. J S.1-2.

<sup>27</sup> Vgl. Europäische Kommission: Bericht der Kommission an das Europäische Parlament und den Rat, in: EUR-LEX, 27.11.2012, [online] <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2012:0699:FIN:DE:PDF> [10.12.2020], S.8.

<sup>28</sup> Vgl. Pfister/Venetz, 2009 S. 439.

<sup>29</sup> Pfister/Venetz, 2009, S. 439.

„XBRL besteht [...] nicht nur aus reiner Datentechnik, sondern auch – und gerade – aus einer großen Portion Sachwissen aus den Bereichen Betriebswirtschaft und Rechnungswesen“<sup>30</sup>

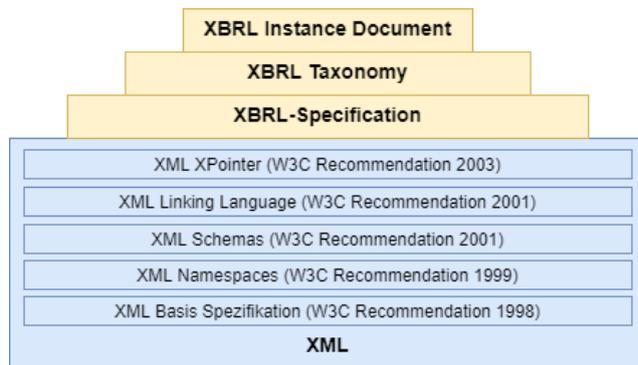


Abbildung 2: Die Schichten von XBRL<sup>31</sup>

Die einzelnen Schichten werden in den folgenden Kapiteln weiter ausgeführt.

## 7.6 XML-Schicht

Ein grobes Verständnis von XML wird für diese Arbeit vorausgesetzt. Im Folgenden sollen aber zwei fortgeschrittenere XML-Konzepte kurz beleuchtet werden, XML Namespaces und die XML Linking Language. Diese sind essenziell für das spätere Verständnis von XBRL.

### 7.6.1 Namespaces

Die in einem XML-Dokument verwendeten Elemente müssen immer einheitlich sein. Es darf keine Elemente mit demselben Namen geben, die eine andere Bedeutung haben. Wenn man das XML-Schema, bzw. die DTD (Dokument Type Definition) selbst entwirft, ist dies einfach umzusetzen. Sobald man aber verschiedenste XML-Schemata von Dritten importiert, kann dies schnell zu einem Problem werden, da man hier keinen Einfluss auf die Benennung der Elemente hat und die Überschneidungsfreiheit somit nicht unbedingt gegeben ist.

Um dieses Problem zu lösen, werden **Namespaces** und deren **Präfixe** verwendet. Mit ihnen können die Elementnamen wieder eindeutig gemacht werden. Außerdem können mithilfe von Namespaces Elemente innerhalb des XML-Dokumentes gruppiert werden.<sup>32</sup>

Das Codebeispiel 1 soll die Anwendung von XML-Namespaces weiter verdeutlichen. Hier wird ein mögliches Datenmodell für die Speicherung von Büchern mit dazugehörigen Hörbüchern gezeigt. Damit die Informationen des Buches von denen des Hörbuches abgrenzbar sind, wird der Namespace „`www.example.com/book`“ für das Buch (Zeile 2) und „`www.example.com/audioBook`“ für das Hörbuch (Zeile 3) definiert.

<sup>30</sup> Flickinger, 2013, S. 24.

<sup>31</sup> Eigene Darstellung nach Flickinger, 2013, S. 24.

<sup>32</sup> Vgl. Debreceny et al., 2009, S. 24.

Anschließend werden mit dem „xmlns“-Attribut (was für „XML-Namespace“ steht) den Namespaces ein Präfix zugewiesen, welcher dann bei der folgenden Benutzung des Elementes verwendet wird. Durch den Namespace ist es nun möglich, die XML-Elemente des Buches von denen des Hörbuches zu unterscheiden. Außerdem ermöglicht der Namespace, die beiden XML-Elemente mit dem Namen „title“ auseinanderzuhalten.

```

1 <item-list
2   xmlns:book="www.example.com/book"
3   xmlns:audio="www.example.com/audioBook">
4   <item>
5     <book:title>Harry Potter und der Stein der Weisen</book:title>
6     <audio:title>Harry Potter 1 gespr. von Rufus Beck</audio:title>
7     <book:author>J.K.Rowling</book:author>
8     <audio:speaker>Rufus Beck</audio:speaker>
9     <category>Fiction</category>
10  </item>
11 </item-list>

```

Codebeispiel 1: Setzen von XML Namespaces

*Ausblick:*

*Während in Codebeispiel 1 die Schemata, welche Buch und Audiobuch beschreiben, mit XML-Namespaces separiert wurden, werden in XBRL verschiedene Taxonomien separiert. So kann beispielsweise eine Taxonomie, die alle finanziellen Konzepte beschreibt und eine andere Taxonomie, die sich auf die Stammdaten des Unternehmens beschränkt importiert werden. Damit die Konzepte der beiden Taxonomien im Instance Document auseinandergelassen werden können, weist man ihnen unterschiedliche Namespaces und Präfixe zu.*

## 7.6.2 XLink

Weit verbreitet bei der Verlinkung von Ressourcen im Web ist das „href“-Attribut. Dieses ermöglicht eine einfache unidirektionale Verlinkung einer anderen Ressource und wird beispielsweise bei dem Anchor Tag „<a>“ in HTML verwendet, um eine Verlinkung auf eine andere Webadresse zu erstellen. Allerdings kommt das „href“-Attribut bei komplexeren Problemstellungen schnell an seine Grenzen, weswegen das World Wide Web Consortium (W3C) mit der **XML Linking Language (XLink)** eine zusätzliche Spezifikation entwickelt hat.

Mit XLink ist die Verlinkung von mehreren Elementen innerhalb sowie zwischen XML-Dokumenten möglich. Dabei können logische Zusammenhänge zwischen zwei oder mehreren XML-Elementen modelliert werden. Somit können neben den bereits bekannten unidirektionalen Verlinkungen auch komplexere Link-Strukturen erstellt werden. Unidirektionale Verlinkungen zwischen zwei Ressourcen werden als **einfache Verlinkung** bezeichnet (`xlink:type="simple"`), die Verlinkung zu einer beliebigen Anzahl an Ressourcen wird als **erweiterte Verlinkung** (`xlink:type="extended"`) bezeichnet.<sup>33</sup>

Im XBRL-Standard werden primär erweiterte Verlinkungen verwendet, daher werden diese im Folgenden genauer erläutert.

<sup>33</sup> Vgl. Debreceny et al., 2009, S. 29-30.

Beim Erstellen eines erweiterten Links macht XLink bezüglich der Namensgebung des Link-Elementes keine Vorgaben. In Codebeispiel 2 bekam es den Namen „studentLink“. Wichtig ist hierbei nur das Attribut „xlink:type“, welches die Art der Verlinkung angibt. Nach dem Erstellen des Link-Elementes müssen zuerst alle externen Elemente, welche mit dem Link-Element verknüpft werden sollen, referenziert werden. Dies wird mit dem sogenannten **Locator** gemacht. Im Codebeispiel 2 wird so die externe Ressource mit dem Namen „student123“, welche in der Schemadatei „students.xsd“ definiert ist, referenziert (Zeile 3 - 4).

Als Nächstes muss die Information, welche mit dem Element verknüpft werden soll, definiert werden. Dieses kann entweder **lokal** (in derselben Datei) erstellt werden oder **remote** (in einer anderen Datei) referenziert werden. Der erste Fall wird mit Zeile 6-10 im Codebeispiel 2 aufgezeigt. Hier werden Vorname und Nachname in separaten lokalen Ressourcen codiert. Über den Wert „ressource“ des Attributes „xlink:type“ erkennt man, dass es sich hier um eine lokal definierte Ressource handelt. Der zweite Fall wird in Zeile 12-13 gezeigt. Hier befindet sich die zu verknüpfende Information in einer anderen Datei, weswegen sie mit einem **Locator** referenziert werden muss.

Nachdem nun klar ist, welche Elemente und Informationen bei der Verknüpfung eine Rolle spielen, muss noch definiert werden, wie diese verknüpft werden sollen.

Hierfür werden die sogenannten **Arcs** verwendet. Diese werden in den Zeilen 15-16 gezeigt und verbinden mit den „from“ und „to“ Attributen zwei XLink Elemente. Die Elemente werden dabei über das „xlink:label“ Attribut identifiziert. Es ist dabei möglich, dass ein Arc-Element über das Label mehrere Elemente verknüpft, wie das bei den zwei Ressourcen im Beispiel (Zeile 6 - 10) der Fall ist, die beide das Label „local“ haben.

```

1  <studentLink xlink:type="extended">
2
3  <locator xlink:type="locator" xlink:label="remote1"
4    xlink:href="/students.xsd#student123"/>
5
6  <resource xlink:type="resource" xlink:label="local"
7    xlink:role="first-name">Peter</resource>
8
9  <resource xlink:type="resource" xlink:label="local"
10   xlink:role="last-name">Smith</resource>
11
12 <locator xlink:type="locator" xlink:label="remote2"
13   xlink:href="/grades-123.xsd"/>
14
15 <go xlink:type="arc" xlink:from="local" xlink:to="remote1"/>
16 <go xlink:type="arc" xlink:from="remote2" xlink:to="remote1"/>
17
18 </studentLink>

```

Codebeispiel 2: *Beispiel eines erweiterten XLinks*

## 7.7 XBRL-Spezifikation und XBRL-Taxonomien

Da XBRL auf XML basiert, erbt XBRL alle Eigenschaften von XML. Zudem erweitert die XBRL-Spezifikation XML um einige Konstrukte, welche sich für das Übertragen von Finanzdaten eignen.<sup>34</sup> XBRL kann man grob in drei Elemente aufteilen: die XBRL-Spezifikation, die XBRL-Taxonomien und die *Instance Documents*.<sup>35</sup> Die XBRL-Spezifikation gibt einen Leitfaden für die Erstellung von Taxonomien und *Instance Documents*. Zudem definiert sie hilfreiche Elemente und Attribute, welche bei der Erstellung, Übertragung, Validation und Analyse der XBRL Reporte benutzt werden können.<sup>36</sup>

Die Taxonomie definiert eine Liste an Konzepten, welche dann im Instance Dokument benutzt werden können und stellt die Integrität und syntaktische Richtigkeit der Daten sicher.

„Es hat sich bewährt, für jedes Rechnungslegungssystem – also etwa IFRS, deutsches Bilanzrecht nach HGB, US-GAAP – eine eigene Taxonomie anzulegen.“<sup>37</sup> Es ist aber auch möglich, von einer anderen Taxonomie zu erben und somit weitere Konzepte hinzuzufügen oder zu verbieten, dazu später mehr in Kapitel 7.7.3 Erweiterbarkeit von Taxonomien.

Die bekanntesten Taxonomien werden im Folgenden kurz erwähnt:

Die **HGB Taxonomie (Handelsgesetzbuch)** wird für das vorher erwähnte Projekt „E-Bilanz“ verwendet.<sup>38</sup> Sie wurde vor allem für die steuerrechtliche Offenlegung entwickelt<sup>39</sup> und besteht aus einem Kernmodul, mehreren branchenspezifischen Modulen und dem GCD-Modul. GCD steht dabei für „Global Common Data/Document“ und bezeichnet das Stammdatenmodul. Dieses enthält Konzepte mit denen die Stammdaten eines Unternehmens gekennzeichnet werden können.<sup>40</sup>

Die **IFRS Taxonomie (International Financial Reporting Standards)** wird von der IFRS Foundation herausgegeben und findet vor allem in der EU Anwendung. Auch bei dem neueren Regulierungsstandard ESEF wird die IFRS Taxonomie verwendet.

Die **US-GAAP Taxonomie (United States Generally Accepted Accounting Principles)** wird von der SEC für das Melden von Quartals- und Jahresberichten für amerikanische Unternehmen vorgeschrieben.

<sup>34</sup> Vgl. Flickinger, 2013, S. 27.

<sup>35</sup> Vgl. Debreceny et al., 2009, S. 38.

<sup>36</sup> Vgl. Wu/Vasarhelyi, 2004, S. 81.

<sup>37</sup> Flickinger, 2013, S. 29.

<sup>38</sup> Vgl. Althoff et al., 2011, S. 13.

<sup>39</sup> Vgl. Althoff et al., 2011, S. 86.

<sup>40</sup> Vgl. XBRL Deutschland e.V.: E-Bilanz / HGB-Taxonomie Version 6.0, in: XBRL DE, [online] <https://de.xbrl.org/taxonomien/e-bilanz-hgb-taxonomie-version-6-0/> [01.01.2021].

### 7.7.1 Taxonomie-Schema

Das Taxonomie-Schema ist das Herzstück der Taxonomie und definiert die Konzepte der Taxonomie. Diese Konzepte stellen die kleinste Informationseinheit eines XBRL-Berichtes dar.<sup>41</sup> Codebeispiel 3 zeigt die Definition eines Konzeptes in dem Taxonomie-Schema. Hier wurde als Beispiel die Definition des Konzeptes „Cash and Cash Equivalents“ (zu Deutsch „Zahlungsmittel und Zahlungsmitteläquivalente“) innerhalb der IFRS Taxonomie aufgezeigt. Bei der Definition des Konzeptes werden erstmal nur der Name und einige andere Eigenschaften definiert. Wichtig hierbei ist vor allem, dass innerhalb der Taxonomie der Name eines Konzeptes nur zu diesem zugeordnet ist und kein weiteres Konzept mit demselben Namen existiert, da diese später im *Instance Document* auch über den Namen referenziert werden.

Mit dem Attribut „type“ (Zeile 6) kann der dem Konzept zugeordnete Datentyp festgelegt werden. Hierbei wird beispielsweise unterschieden, ob es sich um ein Datum, eine Zeichenkette oder um einen monetären Wert handelt.<sup>42</sup>

Weitere erwähnenswerte Attribute sind die XBRL Attribute „xbrli:balance“ und „xbrli:periodType“ (Zeile 7 und 8). Mithilfe des Attributes „xbrli:balance“ kann beschrieben werden, ob das Konzept einen debitorischen oder kreditorischen Charakter besitzt. Das Attribut „xbrli:periodType“ gibt an, ob es sich um ein zeitraumbezogenes Konzept handelt (z.B. Gewinn pro Aktie in der GuV) oder um ein Stichtagswert (z.B. Vermögenswerte in der Bilanz).<sup>43</sup> Die Werte hierfür wären dann entsprechend „duration“ oder „instant“.

```

1   <xsd:element
2       id="ifrs-full_CashAndCashEquivalents"
3       name="CashAndCashEquivalents"
4       nillable="true"
5       substitutionGroup="xbrli:item"
6       type="xbrli:monetaryItemType"
7       xbrli:balance="debit"
8       xbrli:periodType="instant"/>

```

Codebeispiel 3: Deklaration eines Konzeptes am Beispiel der IFRS Taxonomie<sup>44</sup>

<sup>41</sup> Vgl. Althoff et al., 2011, S. 53.

<sup>42</sup> Vgl. Althoff et al., 2011, S. 54.

<sup>43</sup> Vgl. Althoff et al., 2011, S. 54.

<sup>44</sup> Entnommen aus der IFRS Taxonomie von 2020: [http://xbrl.ifrs.org/taxonomy/2020-03-16/full\\_ifrs/full\\_ifrs-cor\\_2020-03-16.xsd](http://xbrl.ifrs.org/taxonomy/2020-03-16/full_ifrs/full_ifrs-cor_2020-03-16.xsd) [12.12.2020].

## 7.7.2 Taxonomie-Linkbases

*Linkbases* sind einzelne XML-Dateien die Struktur in die Konzepte bringen und diese mit weiteren Informationen verknüpfen. Diese Informationen können beispielsweise nutzerfreundliche Bezeichnungen oder Referenzen zu maßgebender Literatur sein. Die *Linkbases* werden im Taxonomie-Schema importiert. Codebeispiel 4 zeigt, wie die *Label Linkbase* mit den deutschen Labels in die ESEF Kern-Taxonomie eingebunden wird. Für die Einbindung wird eine einfache unidirektionale XLink-Verknüpfung verwendet.

```

1 <link:linkbaseRef
2     xlink:type="simple"
3     xlink:href="esef_cor-lab-de.xml"
4     xlink:role="http://www.xbrl.org/2003/role/labelLinkbaseRef"
5     xlink:arcrole=http://www.w3.org/1999/xlink/
      properties/linkbase/>

```

Codebeispiel 4: Einbettung einer Linkbase in ein Taxonomie-Schema<sup>45</sup>

Debreceny et al. teilen in ihrem Buch „XBRL for Interactive Data“ die verschiedenen *Linkbase*-Typen in drei Kategorien ein: „

- Relation linkbases (calculation, definition, and presentation) that manage the relations between taxonomy concepts
- Label linkbases that associate taxonomy concepts with text labels defined in various languages
- Reference linkbases that connect concepts with authoritative literature

„<sup>46</sup>

Alternativ könnte man die beiden Kategorien „*Label Linkbases*“ und „*Reference Linkbases*“ aber auch zu einer Kategorie „*Ressourcen Linkbases*“ zusammenlegen, da diese sich strukturell sehr ähneln.

Zusätzlich zu den bereits genannten *Linkbases* gibt es noch die *Formular Linkbase*. Mit ihr können komplexere mathematische Beziehungen zwischen mehreren Elementen dargestellt werden. Da die aktuelle Basis Spezifikation (XBRL Spezifikation 2.1) *Formular Linkbases* aber nicht unterstützt, wird sie bei der genaueren Betrachtung außen vor gelassen.<sup>47</sup>

<sup>45</sup> Entnommen aus der ESEF Taxonomie von 2020: [http://www.esma.europa.eu/taxonomy/2020-03-16/esef\\_cor.xsd](http://www.esma.europa.eu/taxonomy/2020-03-16/esef_cor.xsd) [13.12.2020].

<sup>46</sup> Debreceny et al., 2009, S. 58.

<sup>47</sup> Debreceny et al., 2009, S. 189.

### 7.7.2.1 Relation Linkbases

Mithilfe von *Relation Linkbases* ist es möglich, hierarchische Strukturen zwischen den Konzepten der Taxonomie darzustellen. Diese hierarchischen Strukturen werden über die im vorherigen Kapitel „7.6.2 XLink“ beschriebenen erweiterten Verlinkungen erstellt. Die *Locators* referenzieren hier auf die Konzepte der Taxonomie, welche im Taxonomie-Schema deklariert sind. Diese *Locators* werden dann über *Arcs* in Verbindung gestellt. Im XLink Jargon ausgedrückt, handelt es sich hier also um die Verlinkung von zwei remote Ressourcen, wobei die remote Ressourcen Konzepte der Taxonomie sind. In dem Beispiel in Abbildung 3 wird ein Locator, welcher auf das Konzept „Assets“ (Vermögenswerte) verweist, über zwei *Arcs* mit den hierarchisch unterliegenden Konzepten „Current Assets“ (Umlaufvermögen) und „Non-Current Assets“ (Anlagevermögen) verbunden.

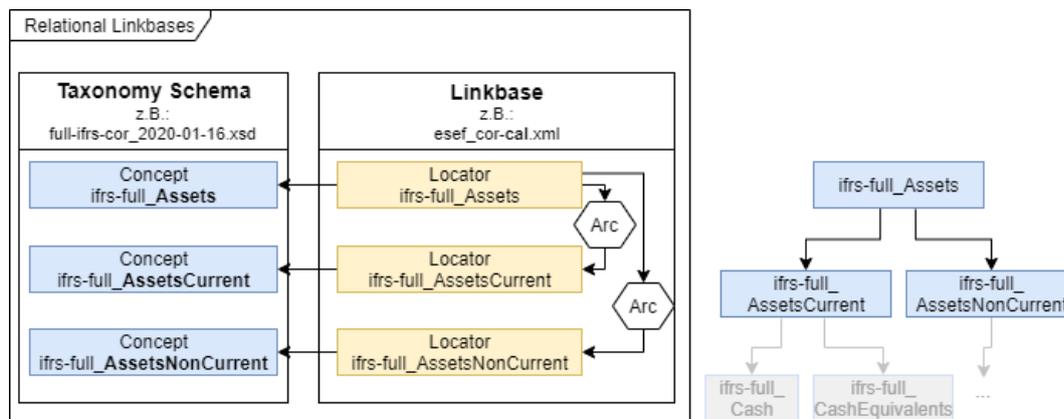


Abbildung 3: Strukturelle Darstellung einer Referenz Linkbase<sup>48</sup>

Codebeispiel 5 zeigt anhand eines Ausschnittes aus der *Calculation Linkbase* der ESEF Taxonomie solch eine hierarchische Verlinkung zwischen zwei Elementen.

Der *Locator* mit dem Label „loc\_8“ referenziert das Konzept „Assets“ aus dem Taxonomie-Schema der IFRS Taxonomie (Zeile 1 - 4). Parallel dazu referenziert der *Locator* mit dem Label „loc\_11“ das Konzept „Current Assets“ (Zeile 6 - 9).

Danach wird in Zeile 11 bis 17 ein *Arc*-Element definiert, welches von dem *Locator* „loc\_8“ auf den *Locator* „loc\_11“ zeigt. Hierdurch wird das Konzept „Assets“, welches durch den *Locator* „loc\_8“ referenziert wird, hierarchisch über das Konzept „Current Assets“ gestellt.

```

1  <link:loc
2    xlink:href="[...]ifrs-cor_2019-03-27.xsd#ifrs-full_Assets"
3    xlink:label="loc_8"
4    xlink:type="locator"/>
5
6  <link:loc
7    xlink:href="[...]ifrs-cor_2019-03-27.xsd#ifrs-full_CurrentAssets
8    xlink:label="loc_11"
9    xlink:type="locator"/>
10
11 <link:calculationArc

```

<sup>48</sup> Eigene Darstellung in Anlehnung an Vgl. Debreceny et al., 2009, S. 59.

```

12     order="20.0"
13     weight="1"
14     xlink:arcrole="http://www.xbrl.org/2003/arcrole/summation-item"
15     xlink:from="loc_8"
16     xlink:to="loc_11"
17     xlink:type="arc"/>

```

Codebeispiel 5: Verlinkung von Konzepten in der Calculation Linkbase<sup>49</sup>

Die in einer *Relation Linkbase* erstellten Hierarchien können je nach *Linkbase*-Typ unterschiedliche Bedeutungen haben. Im Folgenden soll auf diese verschiedenen Typen weiter eingegangen werden.

**Die Calculation Linkbase** definiert einfache rechnerische Beziehungen zwischen einzelnen Konzepten. Hierbei werden meistens Aufsummierungen oder Subtraktionen von Positionen der im Finanzbericht darunterliegenden Gliederungsebene gebildet.<sup>50</sup>

„So entspricht nach HGB zum Beispiel die Position ‚Vorräte‘ der Summe der Posten ‚Roh-, Hilfs- und Betriebsstoffe‘ bis ‚geleistete Anzahlungen‘.“<sup>51</sup>

Um die rechnerischen Beziehungen darzustellen besitzt ein *Calculation-Arc* das Attribut „weight“. Dies ist der Faktor, mit dem das referenzierte Konzept aufsummiert wird. Um eine einfache Subtraktion darzustellen, müsste das Attribut „weight“ den Wert „-1“ besitzen.<sup>52</sup>

Mithilfe der *Calculation Linkbase* kann die rechnerische Korrektheit des Berichtes später automatisch validiert werden. Fehlende Summen- oder Saldenposten können somit sofort erkannt werden, wodurch die Qualität der Berichterstattung sichergestellt werden kann.<sup>53</sup>

**Die Presentation Linkbase** beschreibt durch ihre Hierarchie die Reihenfolge, in der die Konzepte der Taxonomie angeordnet werden sollen.<sup>54</sup> So kann beispielsweise der nach §266 HGB vorgeschriebene Aufbau einer Bilanz mit einer *Presentation Linkbase* vorgegeben und validiert werden.

Codebeispiel 6 enthält ein mögliches Arc-Element einer *Presentation Linkbase*. Dieses zeigt von dem *Locator*, der das Konzept „Vorräte“ referenziert, zu dem *Locator*, der das Konzept „Roh-, Hilfs- und Betriebsstoffe“ referenziert. Das gleiche Prinzip gilt für das zweite Arc-Element, welches das Konzept „Unfertige Erzeugnisse und Leistungen“ unter dem Konzept „Vorräte“ einordnet. Wichtig ist auch das Attribut „order“, welches die Reihenfolge der Konzepte angibt.

```

1     <link:presentationArc
2         order="1"
3         xlink:arcrole="http://www.xbrl.org/2003/arcrole/parent-child
4         xlink:from="loc_Vorräte"

```

<sup>49</sup> Entnommen aus der ESEF Taxonomie von 2019: [https://www.esma.europa.eu/taxonomy/2019-03-27/esef\\_all-cal.xml](https://www.esma.europa.eu/taxonomy/2019-03-27/esef_all-cal.xml) [13.12.2020].

<sup>50</sup> Vgl. Althoff et al., 2011, S. 55.

<sup>51</sup> Flickinger, 2013, S. 37.

<sup>52</sup> Vgl. Flickinger, 2013, S. 38.

<sup>53</sup> Vgl. Althoff et al., 2011, S. 55.

<sup>54</sup> Vgl. Debreceny et al., 2009, S. 60.

```

5     xlink:to="loc_Roh_Hilfs_Betriebsstoffe"
6     xlink:type="arc"/>
7
8     <link:presentationArc
9         order="2"
10        xlink:arcrole="http://www.xbrl.org/2003/arcrole/parent-child
11        xlink:from="loc_Vorräte"
12        xlink:to="loc_unfertige_Erzeugnisse_Leistungen"
13        xlink:type="arc"/>

```

Codebeispiel 6: *Arcs in der Presentation Linkbase*

Die *Locators* wurden in diesem Beispiel weggelassen, da diese in allen *Linkbases* identisch ausschauen (siehe Codebeispiel 5).

Abbildung 4 zeigt noch mal die Struktur auf, welche von der Verlinkung in Codebeispiel 6 erreicht wird.

B.	Umlaufvermögen:
	I. Vorräte:
	1. Roh-, Hilfs- und Betriebsstoffe;
	2. unfertige Erzeugnisse, unfertige Leistungen;
	3. fertige Erzeugnisse und Waren;
	4. geleistete Anzahlungen;

Abbildung 4: *Ausschnitt aus §266 Absatz 2 HGB*

**Die Definition Linkbase** erlaubt es, verschiedene andere logische Zusammenhänge zwischen Konzepten zu erstellen.<sup>55</sup> Beispielsweise kann mit dem Zusammenhang „essence-alias“ betont werden, dass zwei Konzepte denselben oder einen sehr ähnlichen Sachverhalt abdecken. Ein Beispiel hierfür ist der Zusammenhang zwischen den Abschreibungen des Geschäftsjahres aus der GuV-Rechnung und die Abschreibungen des Geschäftsjahres im Anlagespiegel.<sup>56</sup>

### 7.7.2.2 Ressource Linkbases

Im Gegenzug zu den *Relation Linkbases*, welche Verbindungen zwischen verschiedenen Konzepten erstellen, verbinden *Ressource Linkbases* Konzepte mit dazugehörigen Ressourcen. Diese Ressourcen können beispielsweise verschiedene Arten von Bezeichnungen für die Konzepte sein. Im XLink Jargon ausgedrückt wird hier also ein remote Element mit mehreren lokalen Elementen über *Arcs* verknüpft.

Abbildung 5 zeigt eine mögliche Struktur einer Label Linkbase. Hier wird das Konzept „Assets“ mit den Bezeichnungen „Vermögenswerte“ und „Summe Vermögenswerte“ verknüpft. Je nach Verwendung des Konzeptes im Finanzbericht kann eines der zwei Bezeichnungen später ausgesucht werden.

<sup>55</sup> Vgl. Debreceny et al., 2009, S. 63

<sup>56</sup> Vgl. Flickinger, 2013, S. 40.

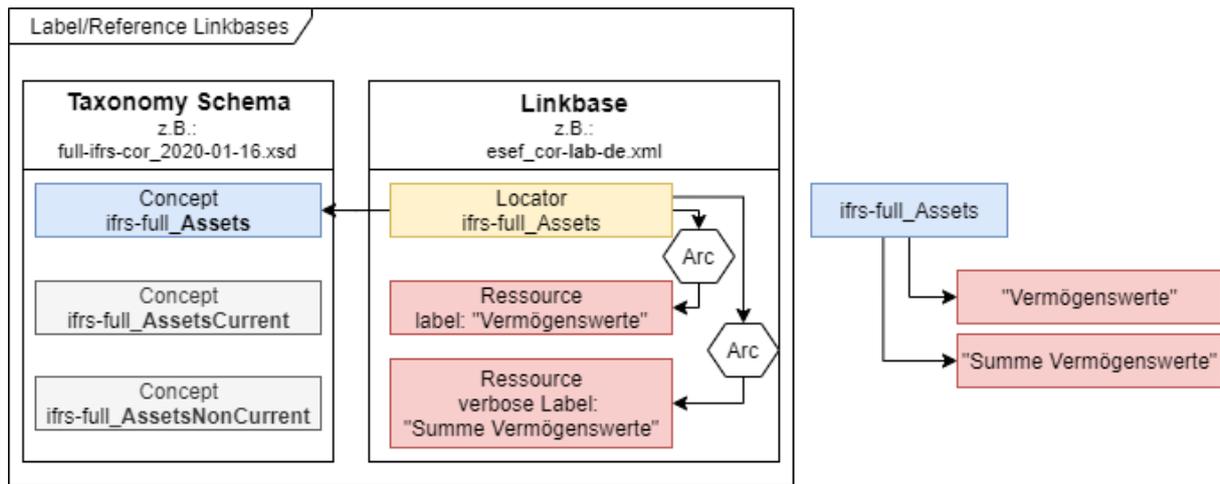


Abbildung 5: Strukturelle Darstellung einer Label Linkbase<sup>57</sup>

Die **Label Linkbase** verknüpft, wie schon eingangs erwähnt, Konzepte mit deren leserfreundlichen Bezeichnungen. Hierbei ist es auch möglich, Bezeichnungen in verschiedenen Sprachen zu verknüpfen. Die Sprache wird dabei über das XML-Attribut „lang“ (Language) angegeben.

Codebeispiel 7 zeigt die Verknüpfung des Konzeptes „Assets“ mit dem deutschen Label „Vermögenswerte“ und einer zusätzlichen Beschreibung in Deutsch.

```

1 <link:loc
2   xlink:href="[...]ifrs-cor_2019-03-27.xsd#ifrs-full_Assets"
3   xlink:label="Assets"
4   xlink:type="locator"/>
5
6 <link:label
7   xlink:type="resource"
8   xlink:role="http://www.xbrl.org/2003/role/label"
9   xml:lang="de"
10  xlink:label="label_Assets">Vermögenswerte</link:label>
11
12 <link:label
13   xlink:type="resource"
14   xlink:role="http://www.xbrl.org/2003/role/documentation"
15   xml:lang="de"
16   xlink:label="documentation_Assets">
17     [...]Der Betrag einer gegenwärtigen wirtschaftlichen Ressource, die
18     vom Unternehmen aufgrund von Ereignissen der Vergangenheit
19     kontrolliert wird. Eine wirtschaftliche Ressource ist ein Recht,
20     aus dem ein wirtschaftlicher Nutzen generiert werden könnte.
21 </link:label>
22
23 <link:labelArc
24   xlink:type="arc"
25   xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
26   xlink:from="Assets"
27   xlink:to="label_Assets"/>

```

<sup>57</sup> Eigene Darstellung in Anlehnung an Debrecey et al., 2009, S. 65.

```

25
26 <link:labelArc
27   xlink:type="arc"
28   xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
29   xlink:from="Assets"
30   xlink:to="documentation_Assets"/>

```

Codebeispiel 7: Verlinkung von Bezeichnungen in der Label Linkbase<sup>58</sup>

Eine der Grundprinzipien von XBRL ist, dass Informationen über den zeitlichen Kontext und die Einheit nicht in der Konzeptdefinition enthalten sein dürfen. Beispielsweise darf kein Konzept definiert werden, welches „Vermögenswerte am Ende des Geschäftsjahres in €“ beschreibt. Stattdessen sollte ein Konzept „Vermögenswerte“ erstellt werden, welches im *Instance Document* dann mit einer Einheit und einem Kontext verbunden werden kann. Um trotzdem die Möglichkeit zu haben, je nach Kontext unterschiedliche Bezeichnungen für die Position zu setzen, ist es in der *Label Linkbase* möglich, Bezeichnungen auch in Abhängigkeit des Zeitraumes zu definieren.<sup>59</sup>

Dies kann durch das „role“-Attribut des Labels (Zeile 8 und 14 des vorherigen Codebeispiels) definiert werden. Beispielsweise kann hier mit dem Wert „periodEndLabel“ eine Bezeichnung definiert werden, die bei Benutzung des Konzeptes am Ende einer Berichtsperiode verwendet werden kann. Das zu benutzende Label wird in der *Presentation Linkbase* angegeben.

Tabelle 1 gibt eine Übersicht über die wichtigsten Label-Arten.

Label Role	Bedeutung
label	Standard Bezeichnung für ein Konzept
terseLabel	Verkürzte Bezeichnung für ein Konzept
verboseLabel	Ausführliche Bezeichnung für ein Konzept
totalLabel	Bezeichnung für ein Konzept, wenn es als Summe anderer Werte benutzt wird
periodStartLabel	Bezeichnung für Werte, die sich auf einen Periodenanfang beziehen
periodEndLabel	Bezeichnung für Werte, die sich auf ein Periodenende beziehen
documentation	Genauere Beschreibung des Konzeptes

Tabelle 1: Übersicht über die wichtigsten Label Role Typen<sup>60</sup>

Die **Referenz Linkbase** kann verwendet werden, um Verlinkungen zwischen Konzepten und Dokumenten außerhalb der XBRL-Technologie zu erstellen.<sup>61</sup> Meistens sind diese externen Dokumente Gesetze oder Richtlinien, welche die Berechnung, Offenlegung oder die Präsentation dieser Konzepte regeln. Codebeispiel 8 zeigt eine Referenz auf die Rechnungslegungsvorschrift IFRS 1 in der Ausgabe 2020, die sich dort auf den Paragraphen IG10 bezieht.

<sup>58</sup> Entnommen aus der ESEF Taxonomie von 2019: [https://www.esma.europa.eu/taxonomy/2019-03-27/esef\\_cor-lab-de.xml](https://www.esma.europa.eu/taxonomy/2019-03-27/esef_cor-lab-de.xml) [13.12.2020].

<sup>59</sup> Vgl. Debreceny et al., 2009, S. 67

<sup>60</sup> Debreceny et al., 2009, S. 68

<sup>61</sup> Vgl. Flickinger, 2013, S. 42.

```

1 <link:reference
2   xlink:type="resource"
3   xlink:role="http://www.xbrl.org/2003/role/disclosureRef"
4   xlink:label="res_1">
5   <ref:Name>IFRS</ref:Name>
6   <ref:Number>1</ref:Number>
7   <ref:IssueDate>2020-01-01</ref:IssueDate>
8   <ref:Paragraph>IG10</ref:Paragraph>
9   <ref:URI>http://eifrs.ifrs.org/eifrs/xifrs-link?type=[..]</ref:URI>
10  <ref:URIDate>2020-01-01</ref:URIDate>
11 </link:reference>

```

Codebeispiel 8: Verlinkung von Literatur in der Referenz Linkbase<sup>62</sup>

Um die Art der Referenz zu beschreiben, wird hier ebenfalls das „role“-Attribut verwendet (Zeile 3). Auch hier gibt es verschiedene Arten von Referenzen. Die wichtigsten werden in Tabelle 2 aufgezeigt.

Referenz Role	Bedeutung
reference	Standard Verweis für ein Konzept
definitionRef	Verweis auf eine genaue Definition des Konzeptes
disclosureRef	Verweis auf eine detaillierte Erläuterung der Offenlegungsanforderungen des Konzeptes
presentationRef	Verweis eine Beschreibung der Anforderungen in Bezug auf Darstellung, Plazierung und Kennzeichnung des Konzeptes in den verschiedenen Berichtstypen
measurementRef	Verweis auf eine Beschreibung der Methoden, die bei der Erfassung dieses Konzeptes verwendet werden müssen
commentaryRef	Verweis auf allgemeine Kommentare und Hinweise zu dem Konzept
exampleRef	Verweis auf Beispiele zu dem referenzierten Konzept

Tabelle 2: Übersicht über die wichtigsten Reference Role Typen<sup>63</sup>

Eine Referenz Linkbase ermöglicht somit, Verlinkungen auf wichtiges Hintergrundwissen zu erstellen, ohne diese Information direkt in XBRL überführen zu müssen. Außerdem wird die gezielte Suche nach Konzepten möglich, die beispielsweise einen Zusammenhang mit einem bestimmten Paragraphen des HGB haben.<sup>64</sup>

### 7.7.3 Erweiterbarkeit von Taxonomien

In der Geschäftsberichterstattung gibt es eine Vielzahl an unterschiedlichen Berichtsanforderungen, die je nach Unternehmensgröße, Industrie und Berichtsart entweder stark variieren können oder auch zu einem großen Teil identisch sein können. Um sich an diese Vielzahl von Anforderungen anpassen zu können, ist ein wichtiges Merkmal des XBRL-Standards die Erweiterbarkeit.<sup>65</sup>

<sup>62</sup> Entnommen aus der IFRS Taxonomie von 2020: [http://xbrl.ifrs.org/taxonomy/2020-03-16/full\\_ifrs/linkbases/ifrs\\_1/ref\\_ifrs\\_1\\_2020-03-16.xml](http://xbrl.ifrs.org/taxonomy/2020-03-16/full_ifrs/linkbases/ifrs_1/ref_ifrs_1_2020-03-16.xml) [20.12.2020].

<sup>63</sup> Debreceny et al., 2009, S. 65 - 66

<sup>64</sup> Vgl. Flickinger, 2013, S. 43.

<sup>65</sup> Vgl. Debreceny et al., 2009, S. 81

Vor allem bei XBRL-Projekten der Europäischen Union wird die Wichtigkeit dieser Eigenschaft deutlich, da hier in vielen Szenarien die EU-weiten Richtlinien in nationale Gesetze umgewandelt werden müssen. So wird beispielsweise die europäische COREP Taxonomie von den einzelnen Mitgliedsstaaten erweitert, um nationale Besonderheiten abdecken zu können.<sup>66</sup> COREP steht dabei für **Common Reporting** und ist eine Taxonomie der Europäischen Bankenaufsichtsbehörde für die Berichterstattung von Kreditinstitutionen und Investmentgesellschaften.

Als erweiternde Taxonomie kann man jede Taxonomie beschreiben, die auf einer anderen Taxonomie aufbaut, diese aber nicht verändert.<sup>67</sup> Die erweiternde Taxonomie erbt sozusagen alle Konzepte und Ressourcen. Hierbei ist es möglich, bestehende Verlinkungen oder Ressourcen zu überschreiben, sowie neue Konzepte, Verlinkungen und Ressourcen hinzuzufügen.<sup>68</sup>

So kann beispielsweise eine erweiternde Taxonomie ein Berechnungsschema für ein Konzept der unterliegenden Taxonomie mit einem neuen Berechnungsschema in der eigenen *Calculation Linkbase* überschreiben.

Generell kann man zwei Strategien bei der Berichterstattung in XBRL unterscheiden. Die **geschlossene Berichterstattung** und die **offene Berichterstattung**. Bei der geschlossenen Berichterstattung muss der Empfänger den genauen Vorgaben der vorgesetzten Taxonomie folgen. Diese Strategie wird gerne in stark regulierten Sektoren wie dem Bankensektor verwendet, da hier ein genaues Einhalten der Rechnungslegungsvorschriften sehr wichtig ist. Bei der offenen Berichterstattung ist es dem berichtenden Unternehmen erlaubt, eine eigene Taxonomie-Erweiterung zu verfassen.<sup>69</sup> Abbildung 6 visualisiert diesen Unterschied nochmals.

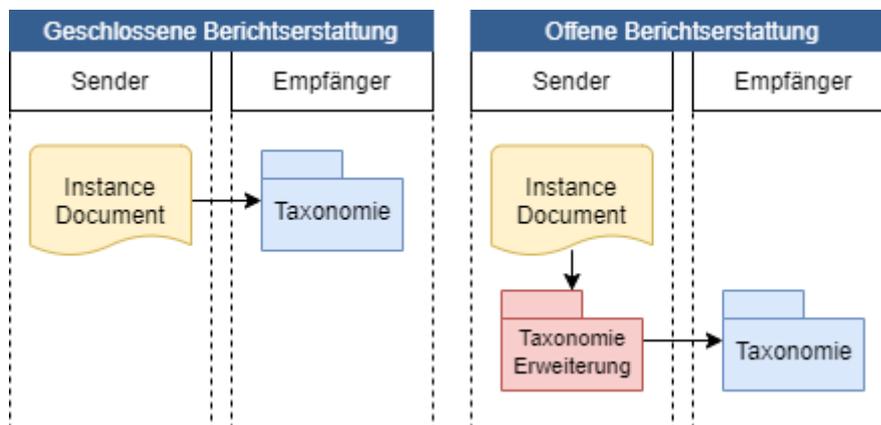


Abbildung 6: Gegenüberstellung der offenen und der geschlossenen Berichterstattung<sup>70</sup>

In der offenen Berichterstattung wird dem Unternehmen durch die Taxonomie-Erweiterung die Erstellung von neuen, unternehmensspezifischen Konzepten ermöglicht.

<sup>66</sup> Vgl. Debreceny et al., 2007, S. 149.

<sup>67</sup> Vgl. Flickinger, 2013, S. 47.

<sup>68</sup> Vgl. Debreceny et al., 2009, S. 85.

<sup>69</sup> Vgl. Debreceny et al., 2009, S. 44.

<sup>70</sup> Eigene Darstellung nach Debreceny et al., 2009, S. 44 – 45.

So kann beispielsweise der Gewinn auf die verschiedenen Unternehmensbereiche oder auf verschiedene Produkte aufgebrochen werden.

#### 7.7.4 Modularisierung von Taxonomien

XBRL Taxonomien können schnell sehr groß und komplex werden. Vor allem wenn eine Taxonomie für mehrere Industrien und über Ländergrenzen hinweg entwickelt werden soll, muss diese systematisch strukturiert werden. Dem Taxonomie Entwickler ist es hier freigestellt, auf wie viele Dateien bzw. XML-Schemata er seine Konzepte aufteilt und wie er die Strukturierung der Verlinkungen und Ressourcen aufbaut.

Häufig gewählte Ansätze sind hier das Separieren von Label Linkbases basierend auf der Sprache, sowie die Gruppierung von logisch zusammenhängenden Konzepten in eine eigene Schemadatei.<sup>71</sup> So besitzt beispielsweise die ESEF Taxonomie für jede in der EU gesprochene Sprache eine eigene Linkbase, welche die Bezeichnungen für die Konzepte in dieser Sprache enthält.

### 7.8 Instance Documents

Die in den letzten Kapiteln behandelten Taxonomien erstellen die Grundstruktur der Berichterstattung in XBRL und setzen die Konzepte meist anhand regulatorischer Richtlinien in Verbindung bzw. verknüpfen sie mit weiteren Ressourcen wie Bezeichnungen.

Das *Instance Document* baut darauf auf und benutzt die in der Taxonomie erstellten Konzepte, um die zu übermittelnden Daten zu kennzeichnen und ihnen somit Bedeutung und Struktur zu geben. Somit verkörpert das *Instance Document* den eigentlichen Geschäftsbericht, beispielsweise den Jahresfinanzabschluss.<sup>72</sup>

Es gibt zwei verschiedene Ausprägungen von *Instance Documents*: das klassische **XBRL Instance Document** und das **iXBRL Instance Document**. Beide Arten basieren auf denselben Elementen, werden aber in unterschiedlichen Dateiformaten verfasst. Während das klassische XBRL *Instance Document* in XML verfasst wird, wird das iXBRL *Instance Document* in XHTML erstellt. Dadurch hat das iXBRL *Instance Document* den Vorteil, dass es auch im Browser betrachtet werden kann.

Beide Ausprägungen bestehen im Wesentlichen aus drei wichtigen Elementen: den **Units (Einheiten)**, den **Contexts (Kontexten)** und den **Facts (Fakten)**. Der Fact enthält die eigentlichen Daten und wird mit einem Konzept aus einer XBRL Taxonomie gekennzeichnet.

Im Folgenden wird auf die XBRL *Instance Documents* eingegangen, da hier die grundlegenden Elemente am einfachsten erkennbar sind. Im anschließenden Kapitel werden dann ein paar signifikante Unterschiede zwischen XBRL und iXBRL behandelt.

<sup>71</sup> Vgl. Goodhand, Mark/Charles Hoffman/Brad Homer/Josef MacDonald/Geoff Shuetrim/Hugh Wallis: XBRL International, in: Financial Reporting Taxonomies Architecture 1.0, 29.01.2005, [online] <http://www.xbrl.org/technical/guidance/old/frta-cr5-2005-01-29.pdf> [22.12.2020], S.1.

<sup>72</sup> Vgl. Flickinger, 2013, S. 51.

### 7.8.1 XBRL Instance Documents

Um das Kennzeichen von Facts in einem XBRL *Instance Document* zu ermöglichen, muss jedes *Instance Document* mindestens eine Taxonomie verknüpfen.<sup>73</sup> Codebeispiel 9 zeigt auf, wie eine Taxonomie in das *Instance Document* eingefügt werden kann.

```
1 <schemaRef type="simple"
2   href="http://www.esma.europa.eu/taxonomy/2019-03-27/esef_cor.xsd"/>
```

Codebeispiel 9: *Einbindung der Taxonomie in das Instance Document*

Nach dem Einbetten einer Taxonomie können jegliche Konzepte, welche in der Taxonomie definiert werden, für das Kennzeichnen der Facts im *Instance Document* verwendet werden. Codebeispiel 10 zeigt einen solchen Fact. Er enthält den Wert „365725000000“ (366 Mrd.) und wird mit dem Konzept „Assets“ aus der US-GAAP-Taxonomie gekennzeichnet. Über dem Präfix „us-gaap“ und dem dazugehörigen Namespace ist die Taxonomie eindeutig definiert.

```
1 <us-gaap:Assets
2   contextRef="FI2018Q4"
3   unitRef="usd"
4   decimals="-6"> 365725000000
5 </us-gaap:Assets>
```

Codebeispiel 10: *Erstellung eines Facts in einem Instance Document*

Der Wert eines Facts muss nicht numerisch sein. Ein Fact kann auch ein Datum oder freien Text beinhalten, je nachdem was bei der Deklaration des Facts im Taxonomie-Schema für ein Datentyp angegeben wurde.

Ein numerischer Fact benötigt allerdings immer eine Einheit. Diese wird über das Attribut „unitRef“ verlinkt und ebenfalls im *Instance Document* deklariert. Codebeispiel 11 zeigt beispielsweise die Einheit US-Dollar (Zeile 1 - 3), welche von dem vorherigen Beispiel referenziert worden ist. Hierbei wird wieder ein Namespace verwendet, um den Begriff „USD“ eindeutig zu machen.<sup>74</sup>

```
1 <xbrli:unit id="usd">
2   <xbrli:measure>iso4217:USD</xbrli:measure>
3 </xbrli:unit>
4
5 <xbrli:unit id="usdPerShare">
6   <xbrli:divide>
7     <xbrli:unitNumerator>
8       <xbrli:measure>iso4217:USD</xbrli:measure>
9     </xbrli:unitNumerator>
10    <xbrli:unitDenominator>
11      <xbrli:measure>xbrli:shares</xbrli:measure>
12    </xbrli:unitDenominator>
13  </xbrli:divide>
14 </xbrli:unit>
```

Codebeispiel 11: *Variationen bei der Erstellung von XBRL-Einheiten*

<sup>73</sup> Vgl. Debreceeny et al., 2009, S. 104.

<sup>74</sup> Vgl. Flickinger, 2013, S. 54.

Ebenfalls ist es möglich zusammengesetzte Einheiten zu erstellen. Die XBRL-Einheit die im vorherigen Codebeispiel Zeile 5-14 deklariert ist, repräsentiert die Einheit US-Dollar pro Aktie und wird bei Konzepten wie beispielsweise „Gewinn pro Aktie“ benötigt.

Um den zeitlichen Bezug eines Facts herzustellen muss jeder Fact einen **Context (Kontext)** referenzieren. Ein Kontext enthält neben dem zeitlichen Bezug außerdem noch eine Identifikationsnummer des berichtenden Unternehmens und die Information, ob sich der Fact auf das gesamte Unternehmen oder nur einen Unternehmensbereich bezieht.<sup>75</sup>

Codebeispiel 12 zeigt die Definition von zwei Kontexten. Der erste Kontext (Zeile 1 - 8) bezieht sich auf ein konkretes Datum, welches beispielsweise ein Bilanzstichtag sein kann. Der zweite Kontext (Zeile 10 - 18) bezieht sich auf einen Zeitraum (01.01.2020 bis 28.09.2020). Solche Zeiträume werden beispielsweise bei der Kapitalflussrechnung verwendet. Der vorher in Codebeispiel 10 gezeigte Fact bezieht sich über das Attribut „contextRef="I2018Q4"“ auf den ersten Kontext mit der entsprechend gleichen Id.

```

1 <xbrli:context id="I2020Q4">
2   <xbrli:entity>
3     <xbrli:identifizier scheme="http://www.sec.gov/CIK">
4       0000320193</xbrli:identifizier>
5   </xbrli:entity>
6   <xbrli:period>
7     <xbrli:instant>2020-09-29</xbrli:instant>
8   </xbrli:period>
9 </xbrli:context>
10 <xbrli:context id="D2020Q3YTD">
11   <xbrli:entity>
12     <xbrli:identifizier scheme="http://www.sec.gov/CIK">
13       0000320193</xbrli:identifizier>
14   </xbrli:entity>
15   <xbrli:period>
16     <xbrli:startDate>2020-01-01</xbrli:startDate>
17     <xbrli:endDate>2020-09-28</xbrli:endDate>
18   </xbrli:period>
19 </xbrli:context>

```

Codebeispiel 12: Variation bei der Erstellung von Kontext

Über das „identifizier“ Element (Zeile 3 und 12) wird definiert, auf welche Firma sich der jeweilige Kontext bezieht. Über das Attribut „scheme“ wird das verwendete Identifikationsnummern-System angegeben. In diesem Beispiel wird der „**Central Index Key**“ (CIK) verwendet, welcher eine eindeutige Identifikationsnummer ist, die von der SEC für jedes, bei der US-Börsenaufsicht registrierte Unternehmen vergeben wird.

<sup>75</sup> Vgl. Debreceny et al., 2009, S. 106.

## 7.8.2 Unterschiede bei iXBRL

Das Rendern von XBRL Berichten war schon immer eines der Schlüsselanforderungen an XBRL.<sup>76</sup> Ein normaler XBRL Bericht hat den Nachteil, dass für einen nutzerfreundlichen Bericht weitere Programme notwendig sind, die mithilfe der Taxonomie, deren Linkbases und externen Stylesheets einen lesbaren Bericht erstellen. Da aber oft ein einheitliches Layout von dem Empfänger gefordert wird, stellt das Rendern der Berichte mit verschiedenen Programmen oft ein Problem dar.<sup>77</sup>

Um eine nutzerfreundliche Ansicht zu gewährleisten, ohne auf den Mehrwert der XBRL-Kennzeichnung zu verzichten, wurde im Jahr 2010 eine erste Spezifikation für das sog. „**inline XBRL**“ (iXBRL) erstellt. Diese macht es möglich, XBRL-Kennzeichnungen in einer HTML Datei vorzunehmen.<sup>78</sup> Gegenüber den XML-Dateien, welche bei dem klassischen XBRL verwendet werden, haben HTML-Dateien den entscheidenden Vorteil, dass diese von einem Endnutzer direkt im Browser lesbar sind. Heutzutage werden ein Großteil der XBRL Berichte in iXBRL verfasst. So stellt die SEC im Zeitraum vom 15.06.2019 bis zum 15.06.2021 alle einzureichenden Jahresabschlüsse und Quartalsberichte auf iXBRL um.<sup>79</sup> Das am 01.01.2020 eingeführte ESEF Reporting basiert ebenfalls auf iXBRL.

Im Folgenden werden kurz die wichtigsten Unterschiede zwischen XBRL und iXBRL aufgezeigt. Wichtig ist nochmals zu betonen, dass XBRL und iXBRL sich nur in dem Instance Dokument unterscheiden und beide auf denselben Taxonomien und Ideen basieren.

Codebeispiel 13 zeigt anhand des vorher verwendeten Facts die Unterschiede zwischen XBRL und iXBRL. Hier fällt sofort auf, dass der Fact in iXBRL mehr Attribute hat als in XBRL. Das lässt sich vor allem darauf zurückführen, dass numerische Werte in XBRL immer als ganze Zahlen ausgeschrieben werden, während sie im iXBRL in verschiedenen Formaten auftreten können. Dies ist wichtig, da die Werte im iXBRL direkt von dem Nutzer im Browser gelesen werden. Um Werte in iXBRL richtig herauslesen zu können, muss zuerst das Attribut „format“ betrachtet werden und anschließend der Wert mithilfe des Attributs „scale“ multipliziert werden.

### XBRL (XML)

```
1 <us-gaap:Assets
2   contextRef="FI2018Q4"
3   unitRef="usd"
4   decimals="-6">
5     365725000000
6 </us-gaap:Assets>
```

### iXBRL (xHTML)

```
1 <ix:nonFraction
2   contextRef="FI2018Q4"
3   unitRef="usd"
4   decimals="-6"
5   format="ixt:numdotdecimal"
7   name="us-gaap:Assets"
8   scale="6">
5     365,725
6 </ix:nonFraction>
```

Codebeispiel 13: Unterschiede bei der Fact-Deklaration zwischen XBRL und iXBRL

<sup>76</sup> Vgl. Debreceny et al., 2009, S. 205.

<sup>77</sup> Vgl. Flickinger, 2013, S. 68.

<sup>78</sup> Vgl. Debreceny et al., 2009, S. 206.

<sup>79</sup> Vgl. U.S. Securities and Exchange Commission: Inline XBRL Filing of Tagged Data, in: SEC Website, 28.06.2018, [online] <https://www.sec.gov/rules/final/2018/33-10514.pdf> [27.11.2020], S. 42

## 7.9 Übersicht über einen vollständigen Bericht

Abbildung 7 zeigt eine Gesamtansicht der zuletzt behandelten Themen. Hier steht im *Instance Document* der Wert „73.156.564“, welcher nun mit weiteren Informationen verknüpft wird. Innerhalb des *Instance Documents* wird die Einheit hinzugefügt und ein zeitlicher Kontext gegeben. Es ist nun also klar, dass es sich um den Wert „73.156.564 USD“ vom 28.09.2020 handelt. Da der Fact mit dem Konzept „Assets“ aus der Taxonomie gekennzeichnet worden ist, ist klar, dass es sich um die Vermögenswerte (nach US-GAAP berechnet) handelt.

Über die Linkbases der Taxonomie können nun weitere Informationen über Darstellung, Berechnung, Bezeichnungen sowie die Einordnung des Konzeptes herausgefunden werden. So kann beispielsweise eine Beschreibung des Konzeptes über die Label Linkbase aufgefunden werden. Zusätzlich ist durch die *Calculation Linkbase* jetzt klar, dass der Wert „73.156.564 USD“ aus der Summe des Umlaufvermögens („Current Assets“) und Anlagevermögen („Non-Current Assets“) entstanden ist.

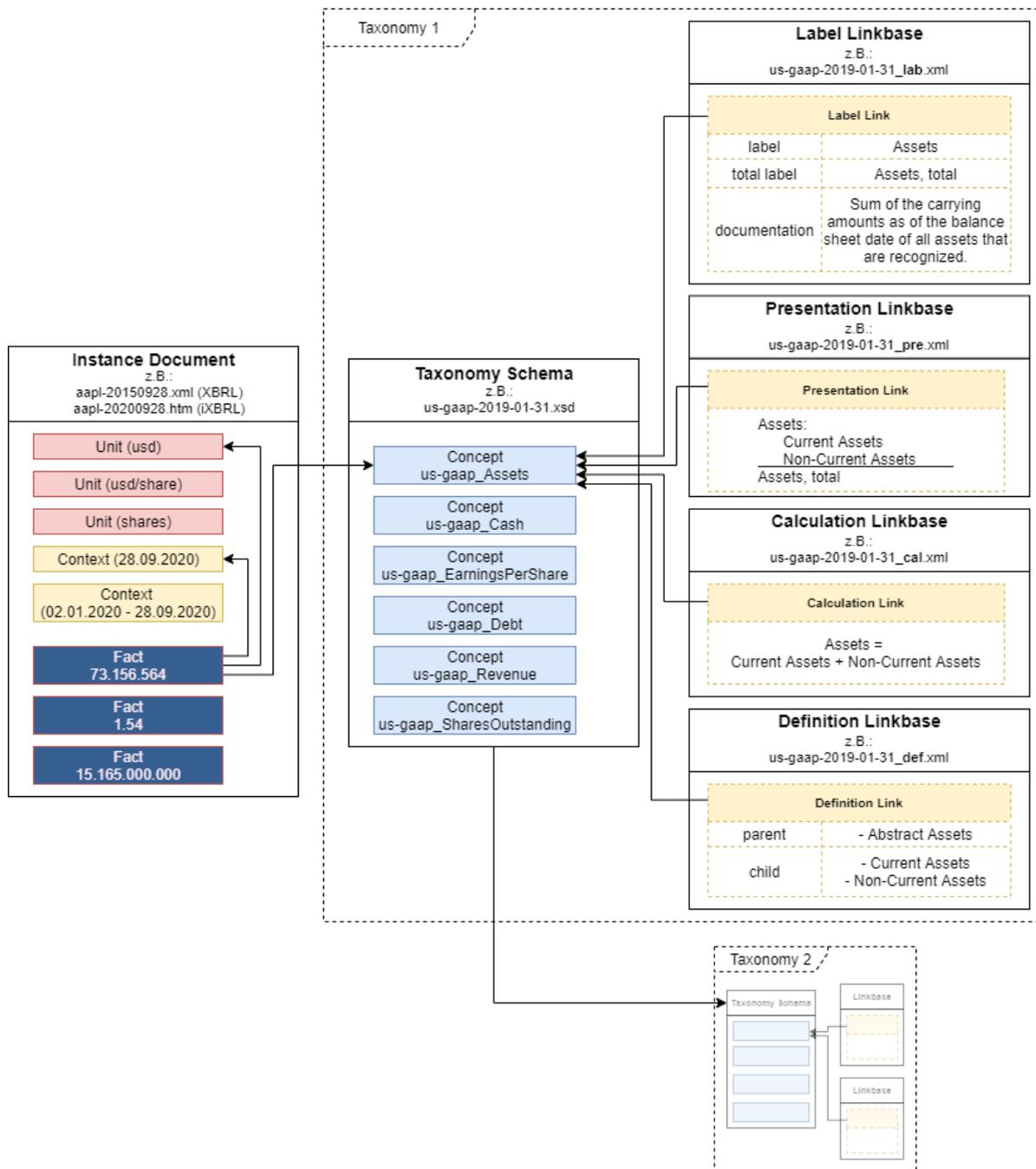


Abbildung 7: Struktur eines Instance Documents mit Taxonomie<sup>80</sup>

<sup>80</sup> Eigene Darstellung in Anlehnung an Debrecey et al., 2009, S. 12.

## 8 Anwendungsbeispiel

### 8.1 Ziele des Projektes

Wie Eingangs bereits erwähnt, wurde mit der Einführung des European Single Electronic Format (ESEF) ab dem 01. Januar 2020 die Erstellung von Jahresberichten im XBRL-Standard für kapitalmarktorientierte Unternehmen innerhalb der EU verpflichtend. Die zugrunde liegende Transparenzrichtlinie, welche die Entwicklung eines elektronischen Berichtsformates gefordert hat, setzte sich als eines der zentralen Ziele die Verbesserung der Zugänglichkeit, Analyse und Vergleichbarkeit von Jahresfinanzberichten für Anleger (Richtlinie 2013/50/EU Artikel 26).

Im Rahmen dieses Anwendungsbeispiels sollen daher vor allem die Hürden, auf welche Anleger bei der maschinellen Analyse von XBRL-Berichten stoßen können, erforscht werden. Dabei soll auch gezeigt werden, wie XBRL-Berichte abgerufen und eingelesen werden können. Zum Schluss soll auf Basis der extrahierten Daten eine beispielhafte Analyse der durchschnittlichen Kurs-Gewinn-Verhältnisse (KGV) für verschiedene Sektoren erstellt werden.

### 8.2 Open Source Lösungen

Für die Erstellung eines Berichtes in XBRL gibt es schon reichlich proprietäre Standardsoftware, die Unternehmen bei der Erfüllung ihrer Berichtspflichten helfen. Allerdings gibt es für die Empfängerseite aktuell nur eine geringe Anzahl an Software Lösungen, welche die Analyse von mehreren Unternehmen ermöglichen. Noch kleiner wird die Teilmenge, wenn hier nach Open Source Lösungen gesucht wird.

Eine dieser wenigen Open Source Lösungen ist das Projekt „Arelle<sup>®</sup>“. Dieses umfasst eine Vielzahl von Features wie das Erstellen, Editieren, Validieren und Rendern von XBRL-Berichten. Außerdem bietet Arelle<sup>®</sup> auch Möglichkeiten Berichte einzulesen und an externe Analyseprogramme weiterzugeben.<sup>81</sup> Allerdings ist das Projekt aufgrund seiner vielen Features sehr komplex und bietet zudem keine einfache Möglichkeit, bestimmte Facts aus XBRL-Berichten auszulesen und z.B. in einer Datenbank abzuspeichern.

Andere nutzbare Open Source Lösungen wurden im Rahmen dieser Arbeit nicht gefunden. Es gibt zwar ein paar Packages (z.B. das „xbrl“ Package im Python Package Index), diese sind aber meistens entweder nie fertig entwickelt worden oder lesen nur einzelne Facts aus dem *Instance Document* ein, ohne die dahinterliegenden Taxonomien anzufassen.

Aus diesen Gründen wurde entschieden einen eigenen XBRL-Parser zu entwickeln.

---

<sup>81</sup> Vgl. Fischer, Herm/Mark V Systems Limited/XBRLSpy Research Inc./Diane Mueller: Enabling Comparability and Data Mining with the Arelle<sup>®</sup> Open Source Unified Model, in: Arelle, 2011, [online] <https://arelle.org/arelle/download/1014/> [02.01.2021].

### 8.3 Aufbau des Projektes

Für die Implementierung der Applikation wurde Python als Programmiersprache gewählt. Python hat den großen Vorteil, dass die Programmiersprache ein sehr großes Ökosystem an externen Bibliotheken besitzt. Vor allem im Bereich Data Science und Web Scraping hat sich Python weit etabliert. Die große Auswahl an externen Bibliotheken wird vor allem bei der späteren Analyse der extrahierten Finanzdaten sehr von Vorteil, da durch die Funktionsunterstützung von Bibliotheken wie „Pandas“, „Numpy“ und „Matplotlib“ die Verarbeitung und Visualisierung der Finanzdaten einfach und schnell implementiert werden kann.

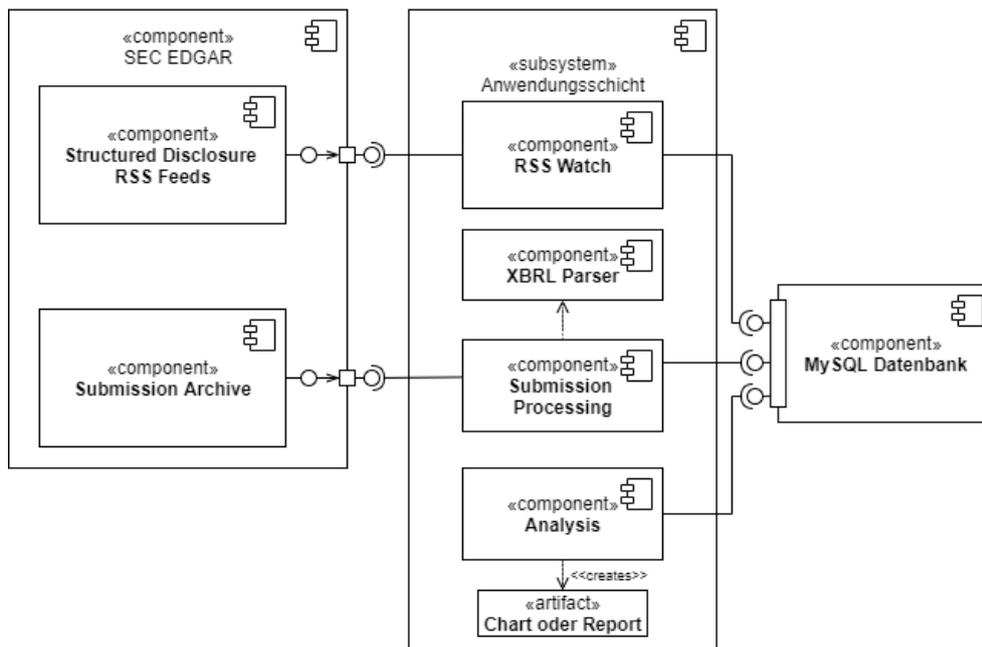
Als Datenbank wurde MySQL gewählt, da MySQL eine der bekanntesten Open Source Datenbanken ist und sich deswegen gut als Referenz einsetzen lässt. Die Datenbank wird einerseits für das Abspeichern von Metainformationen zu XBRL-Einreichungen, sowie für das Ablegen der aus den XBRL-Berichten herausgelesenen Facts benötigt.

Die XBRL-Berichte wurden von der US-Börsenaufsicht SEC abgerufen, da diese die größte Sammlung an öffentlich abrufbaren XBRL-Berichten anbietet. Eigentlich sollten auch Jahresabschlüsse, welche mit der neuen ESEF-Taxonomie erstellt worden sind, in die maschinelle Analyse mit einfließen. Da allerdings zum Zeitpunkt der Verfassung der Arbeit noch keine Jahresabschlüsse für das Geschäftsjahr 2020 vorlagen, konnten diese nicht verwendet werden. Dies ist darauf zurückzuführen, dass die Abgabefrist der Jahresberichte des Geschäftsjahres 2020 zum Zeitpunkt der Verfassung dieser Arbeit noch nicht vergangen ist. Zudem wird aktuell eine Verschiebung der ESEF-Richtlinie um ein Jahr erwogen, um Unternehmen welche aufgrund der Covid-19 Pandemie sowieso schon in Schwierigkeiten geraten sind keine weitere Belastung aufzusetzen.<sup>82</sup> Das Europäische Parlament stimmte der Verschiebung um ein Jahr am 11. Dezember 2020 zu, allerdings gibt es hierzu noch keine gesetzliche Umsetzung auf deutscher Seite.

Abbildung 8 gibt eine Übersicht über die verschiedenen Komponenten und Schnittstellen des Anwendungsbeispiels. Im Rahmen dieser Bachelorarbeit wurden die Anwendungsschicht und die Architektur der Datenbank entwickelt.

---

<sup>82</sup> Vgl. Delivorias, Angelos/Europäisches Parlament: Prospectuses for investors – Simplifying equity-raising during the pandemic, in: Europäisches Parlament, 19.11.2020, [online] [https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/659350/EPRS\\_BRI\(2020\)659350\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/659350/EPRS_BRI(2020)659350_EN.pdf) [10.01.2021].

Abbildung 8: Komponentendiagramm des Projektes<sup>83</sup>

## 8.4 SEC EDGAR

In den USA müssen börsengehandelte Unternehmen regelmäßig finanzielle Offenlegungen bei der US-Börsenaufsicht (SEC) einreichen. Um die Transparenz und Verfügbarkeit dieser Offenlegungen zu verbessern, begann die SEC im Jahre 1983 mit der Entwicklung des **Electronic Data Gathering, Analysis, and Retrieval (EDGAR)** Systems. Mit dem EDGAR-System ist es für jeden möglich, gesetzlich vorgeschriebene Berichte von öffentlichen Firmen kostenlos abzurufen.<sup>84</sup>

Mithilfe von EDGAR will die SEC „die Effizienz und Fairness des Wertpapiermarktes zum Nutzen von Investoren, Unternehmen und der Wirtschaft erhöhen, indem sie den Empfang, die Annahme, die Verbreitung und die Analyse von zeitkritischen Unternehmensinformationen, die bei der Behörde eingereicht werden, beschleunigt“.<sup>85</sup>

Aktuell (Sept. 2020) enthält EDGAR 13,8 Mio. Berichte, wovon ca. 3 Mio. im XBRL Format vorliegen.<sup>86</sup> Diese Berichte werden von der Öffentlichkeit stark nachgefragt. So werden pro Jahr ca. 3 Petabyte an Daten von den Servern der SEC abgerufen.<sup>87</sup>

<sup>83</sup> Eigene Darstellung

<sup>84</sup> Vgl. Gerdes, John: EDGAR-Analyser: automating the analysis of corporate data contained in the SEC's EDGAR database, in: Decision Support Systems, Jg. 2003, Nr. 35, 2003, doi: 10.1016/s0167-9236(02)00096-9, S. 7.

<sup>85</sup> Gerdes, 2003, S. 9.

<sup>86</sup> Die Zahlen sind aus der CSV Datei „EDGAR Filing Counts by FormType“ entnommen und sind auf folgender Website herunterladbar: [https://www.sec.gov/dera/data/dera\\_edgarfilingcounts](https://www.sec.gov/dera/data/dera_edgarfilingcounts), [25.12.2020]

<sup>87</sup> Vgl. Securities and Exchange Commission: About EDGAR, in: U.S. Securities and Exchange Commission, 24.08.2020, [online] <https://www.sec.gov/edgar/about> [27.12.2020].

Für das Anwendungsbeispiel sind vor allem Jahresabschlüsse und Quartalsberichte interessant. Diese kann man anhand der Formularart unterscheiden. Jahresabschlüsse werden unter dem Formulkürzel **10-K** und Quartalsberichte unter **10-Q** geführt.

### 8.4.1 Aufbau der Jahres- und Quartalsberichte

Die amerikanische Börsenaufsicht erlaubt berichtenden Unternehmen eine eigene Taxonomie-Erweiterungen (basierend auf der US-GAAP Taxonomie) zu erstellen. Damit verfolgt sie eine offene Berichtserstattung. Ein durchschnittlicher XBRL-Bericht aus dem EDGAR-System besteht somit meistens aus dem *Instance Document* und einer Taxonomie-Erweiterung. Die Taxonomie-Erweiterung wiederum umfasst ein Taxonomie-schema und die *Linkbases*.

### 8.4.2 Schnittstellen von EDGAR

Zum Abrufen der Berichte stellt die SEC mehrere Möglichkeiten zur Verfügung:

**Company Search:** Ein einfaches webbasiertes Formular, wo der Nutzer einen Firmennamen, ein Ticker Symbol oder ein CIK (Central Index Key - eindeutige Identifikationsnummer, welche von der SEC vergeben wird) eingeben kann und eine Liste von Suchergebnissen präsentiert bekommt.<sup>88</sup> Eignet sich für die manuelle Abfrage eines Berichtes.

**Index Files:** Die SEC stellt auch Index Dateien bereit, in denen alle Berichte für einen bestimmten Zeitraum aufgezeichnet sind. Diese Index Dateien liegen in verschiedenen Sortierungen und für verschiedene Zeiträume vor. Abbildung 9 zeigt einen Ausschnitt aus einer Index Datei welche alle Berichte, die am 30.06.2020 eingereicht worden sind, enthält. Die Berichte wurden hier nach Unternehmensnamen sortiert.

```

1 Description:          Daily Index of EDGAR Dissemination Feed by Company Name
2 Last Data Received:  Jun 30, 2020
3 Comments:            webmaster@sec.gov
4 Anonymous FTP:      ftp://ftp.sec.gov/edgar/
5
6
7 Company Name          Form Type  CIK
8   Date Filed  File Name
9 -----
10 100 Acre Premium Arb SPV I, LP      D          1815739   20200630   edgar/data/1815739/0001815739-20-000001.txt
11 12 Retch Corp                     NT 10-Q/A   1627611   20200630   edgar/data/1627611/0001493152-20-012171.txt
12 12 West Capital Management LP      4          1540531   20200630   edgar/data/1540531/0000905718-20-000674.txt
13 1840 Investors Series LLC          D          1816301   20200630   edgar/data/1816301/0001816301-20-000001.txt
14 1ST SOURCE CORP                    11-K       34782    20200629   edgar/data/34782/0000034782-20-000098.txt
15 1WS Credit Income Fund             N-CSRS     1748680   20200630   edgar/data/1748680/0001398344-20-013042.txt
16 2020 Hybrid Drilling Program, L.P.  D          1816353   20200630   edgar/data/1816353/0001816353-20-000001.txt
17 22nd Century Group, Inc.           4          1347858   20200630   edgar/data/1347858/0001567619-20-012691.txt

```

Abbildung 9: Ausschnitt einer Index Datei des EDGAR Systems<sup>89</sup>

**Directory Browsing:** Wenn die CIK des gesuchten Unternehmens vorliegt, ist es auch möglich eine URL aufzurufen, welche auf einen Ortner mit allen Berichten des Unternehmens verweist. So findet man z.B. unter der URL „<https://www.sec.gov/Archives/edgar/data/320193/>“ alle Berichte des Unternehmens „Apple Inc.“ mit der CIK „320193“.

<sup>88</sup> Siehe <https://www.sec.gov/edgar/searchedgar/companysearch.html>

<sup>89</sup> Entnommen von <https://www.sec.gov/Archives/edgar/daily-index/2020/QTR2/company.20200630.idx>

**Structured Disclosure RSS Feeds:** Eine weitere Schnittstelle des EDGAR Systems sind die von der SEC bereitgestellten RSS-Feeds. Diese werden alle 10 Minuten durch die SEC aktualisiert und werden in zwei Versionen angeboten. Die erste Version der RSS Feeds enthält die letzten 200 Einreichungen, nach Annahmedatum sortiert. Die zweite Version enthält jeweils alle Einreichungen, die in einem bestimmten Monat getätigt worden sind. Im Folgenden Anwendungsbeispiel werden nur die monatlichen RSS Feeds weiter betrachtet.

Ein großer Vorteil der RSS Feeds ist, dass sie neben der URL zu dem XBRL-Bericht auch weitere Daten wie z.B. Unternehmensname, Berichtstyp, Berichtsdatum und Einstellungsdatum beinhalten. Dadurch werden einfache Abfragen wie nach Unternehmensnamen oder Berichtsart ohne vorherige Auswertung des *Instance Documents* möglich. Im Codebeispiel 14 wird ein einzelner, verkürzter Eintrag im RSS Feed zu einen Quartalsbericht von Microsoft gezeigt.

```

1 <item>
2   <title>MICROSOFT CORP (0000789019) (Filer)</title>
3   <link>https://www.sec.gov/Archives/edgar/[...].htm</link>
4   <description>10-Q</description>
5   <edgar:xbrlFiling>
6     <edgar:companyName>MICROSOFT CORP</edgar:companyName>
7     <edgar:formType>10-Q</edgar:formType>
8     <edgar:filingDate>10/27/2020</edgar:filingDate>
9     <edgar:period>20200930</edgar:period>
10    <edgar:assignedSic>7372</edgar:assignedSic>
11    <edgar:fiscalYearEnd>0630</edgar:fiscalYearEnd>
12    <edgar:xbrlFiles>[...]</edgar:xbrlFiles>
13  </edgar:xbrlFiling>
14 </item>

```

Codebeispiel 14: Auszug des monatlichen RSS Feeds von EDGAR<sup>90</sup>

Die Auflistung der einzelnen XBRL Dateien (Zeile 12) wurde zur Vereinfachung weggelassen. Hier handelt es sich nur um eine einfache Auflistung der einzelnen Dateien des Berichtes, jeweils mit Dateigröße, Datei-URL und Dateityp.

## 8.5 Anwendungsschicht

### 8.5.1 Einlesen des Structured Disclosure RSS Feeds von EDGAR

Wie schon vorher erwähnt, werden im Projekt die monatlich aggregierten RSS Feeds verwendet, da diese den Zugriff auf Berichte vom Jahr 2005 bis zum aktuellen Datum ermöglichen und zusätzliche Informationen zu den Berichten bereitstellen. Da aber allein die monatlichen RSS Index Dateien insgesamt auf 1,6 GB Dateigröße kommen (Stand Dez. 2020), müssen diese lokal zwischengespeichert werden, um eine schnelle Abfragezeit zu erreichen und den Datenverkehr einigermaßen gering zu halten. Hierfür wird eine MySQL Datenbank verwendet, welche die gesamten RSS Feeds abspeichert und leicht abfragbar macht. Abbildung 10 zeigt die Struktur dieser Datenbank. In ihr werden alle Informationen abgespeichert, welche auch im RSS Feed von EDGAR verfügbar sind.

<sup>90</sup> Entnommen aus <https://www.sec.gov/Archives/edgar/monthly/xbrlrss-2020-10.xml>, stark abgekürzt

Dadurch ist es möglich, einfach und effizient eine gezielte Teilmenge von Berichten aus EDGAR abzufragen, ohne jedes Mal alle RSS Index Dateien von den SEC Servern herunterladen zu müssen.

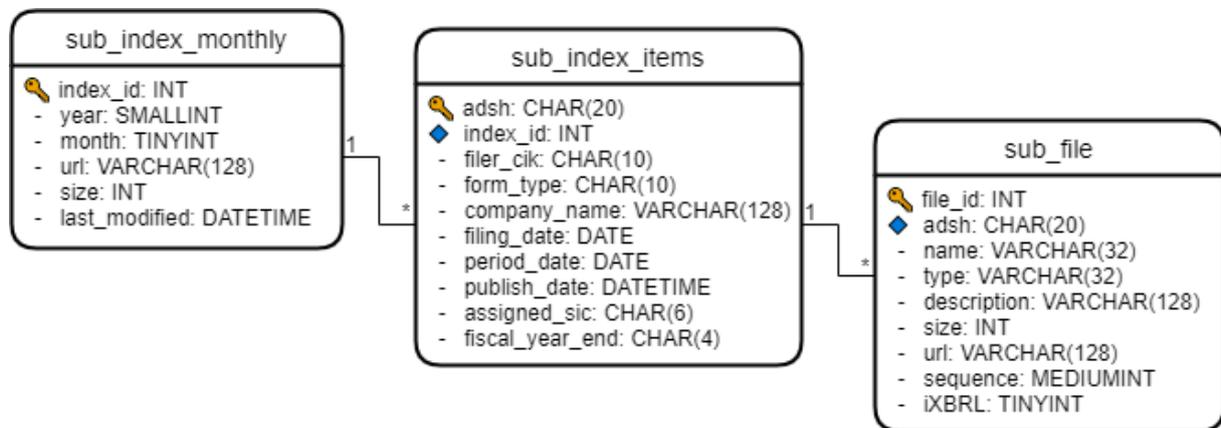


Abbildung 10: Struktur der RSS Index Tabellen<sup>91</sup>

Im Folgenden werden die verschiedenen Tabellen der Datenbank kurz erklärt:

**sub\_index\_monthly:** Die Tabelle „sub\_index\_monthly“ („monthly submission index“) enthält für jede monatliche RSS Index Datei einen Eintrag. Beispielsweise wird für die monatliche RSS Index Datei „xbrlrss-2020-12.xml“ ein Eintrag angelegt. Alle verfügbaren monatlichen RSS Index Dateien können unter <https://www.sec.gov/Archives/edgar/monthly/> betrachtet werden.

**sub\_index\_items:** Die Tabelle „sub\_index\_items“ („submission index items“) speichert die einzelnen eingereichten Berichte mit zusätzlichen Metadaten ab. Hierbei wird jedes Item des RSS Feeds als einzelner Bericht verstanden. Da einige Spalten dieser Tabelle nicht selbstverständlich sind, werden sie in Tabelle 3 weiter erläutert.

Feldname	Erläuterung
<b>adsh</b>	Eindeutige, von der SEC für den Bericht vergebene ID
<b>filer_cik</b>	Der Central Index Key des Unternehmens, welches den Bericht eingereicht hat
<b>company_name</b>	Name des Unternehmens, welches den Bericht eingereicht hat
<b>form_type</b>	Art des Berichtes (10-K, 10-Q, ...)
<b>filing_date</b>	Datum an dem der Bericht bei der SEC eingereicht wurde
<b>period_date</b>	Datum auf welches sich der Bericht bezieht (Bilanzstichtag)
<b>publish_date</b>	Zeitstempel, an dem der Bericht in EDGAR veröffentlicht wurde
<b>assigned_sic</b>	SIC steht für Standard Industrial Classification und ist ein Klassifikationsschema für unterschiedliche Industriezweige. Mit diesem Feld lässt sich das einreichende Unternehmen einer Branche zuweisen
<b>fiscal_year_end</b>	Ende des Geschäftsjahres in der Form MMDD

Tabelle 3: Erläuterung der Spalten der sub\_index\_items Tabelle

**sub\_file:** Die Tabelle „sub\_file“ („submission index file“) speichert die einzelnen Dateien der Einreichung ab. Über die Spalte „type“ ist es möglich, die Art der Datei festzustellen. Dies können beispielsweise das Taxonomie-Schema, das Instance Dokument, verschiedene

<sup>91</sup> Eigene Darstellung

Linkbases oder auch im Bericht eingebettete Grafiken sein. Die Spalte „iXBRL“ gibt an, ob der Bericht im XBRL-Format oder im iXBRL-Format erstellt worden ist.

Damit die lokale Repräsentation des RSS Indexes immer aktuell mit dem auf den SEC Servern vorhandenen RSS Index bleibt, wird jeweils der Zeitstempel der letzten Änderung verglichen. Dafür wird das Feld „last\_modified“ in der Tabelle „sub\_index\_monthly“ mit der zusammenfassenden Index Datei auf den SEC Servern abgeglichen. Diese Index Datei enthält eine JSON Repräsentation des RSS Index-Verzeichnisses und gibt für jeden monatlichen RSS Index einen Zeitstempel der letzten Änderung an.<sup>92</sup>

### 8.5.2 Caching Strategie

Bevor man mit dem Auslesen und Analysieren von Daten beginnt, muss eine Cache Strategie entwickelt werden, da beim Auslesen der XBRL-Berichte sehr viele Dateien anfallen. Für dieses Projekt wurde ein einfacher Disk Cache verwendet, welcher beim ersten Aufruf mit einer URL die Datei herunterlädt und lokal abspeichert. Bei allen weiteren Aufrufen mit derselben URL wird ein Pfad zu der lokal abgespeicherten Datei zurückgegeben. Damit eine im Disk Cache abgespeicherte Datei anhand ihrer URL schnell gefunden werden kann, wird im Cache die URL für die Erstellung des Cache-Pfades verwendet. So wird beispielsweise die Datei mit der URL „<https://www.sec.gov/Archives/edgar/data/320193/000032019318000100/aapl-20180630.xml>“ im Disk Cache unter dem Pfad „D:\cache\www.sec.gov\Archives\edgar\data\320193\000032019318000100\aaapl-20180630.xml“ abgespeichert. Das Stammverzeichnis des Disk Caches (hier „D:\cache“ kann dabei über eine Umgebungsvariable gesteuert werden.

### 8.5.3 Aufbau und Funktionsweise des Parsers

Der für das Projekt verwendete Parser wurde objektorientiert mit Python entwickelt und besteht aus drei großen Modulen: dem **Instance Document Modul**, dem **Taxonomie-Schema Modul** und dem **Linkbase Modul**. Zum Einlesen der Dateien wurde der XML-Parser „ElementTree“ verwendet.

Ziel des XBRL-Parsers ist, die verschiedenen XML-, HTML- und XSD-Dateien einzulesen und in ein logisches Konstrukt von Objektinstanzen zu übertragen. Mit den sich im Arbeitsspeicher befindenden Objektinstanzen können dann einfach und schnell weitere Aktionen getätigt werden. So kann beispielsweise über alle Facts iteriert werden, um Werte, die mit einem bestimmten Konzept gekennzeichnet sind, herauszufiltern. Im Anhang befindet sich die Struktur dieser Objektinstanzen in Form eines Klassendiagramms. Im Folgenden soll der Vorgang des Einlesens der XBRL-Berichte und deren referenzierten Taxonomien anhand eines Aktivitätsdiagramms erklärt werden.

---

<sup>92</sup> Diese Index Datei befindet sich unter <https://www.sec.gov/Archives/edgar/monthly/index.json>

Das Aktivitätsdiagramm wurde allerdings etwas vereinfacht, da das rekursive Aufrufen der „`parse_taxonomy()`“ Methode für importierte Taxonomien weggelassen wurde. Das in Abbildung 11 gezeigte Diagramm stellt daher den Ablauf des Einlesens einer Taxonomie dar, welche keine weitere Taxonomie-Importe hat.

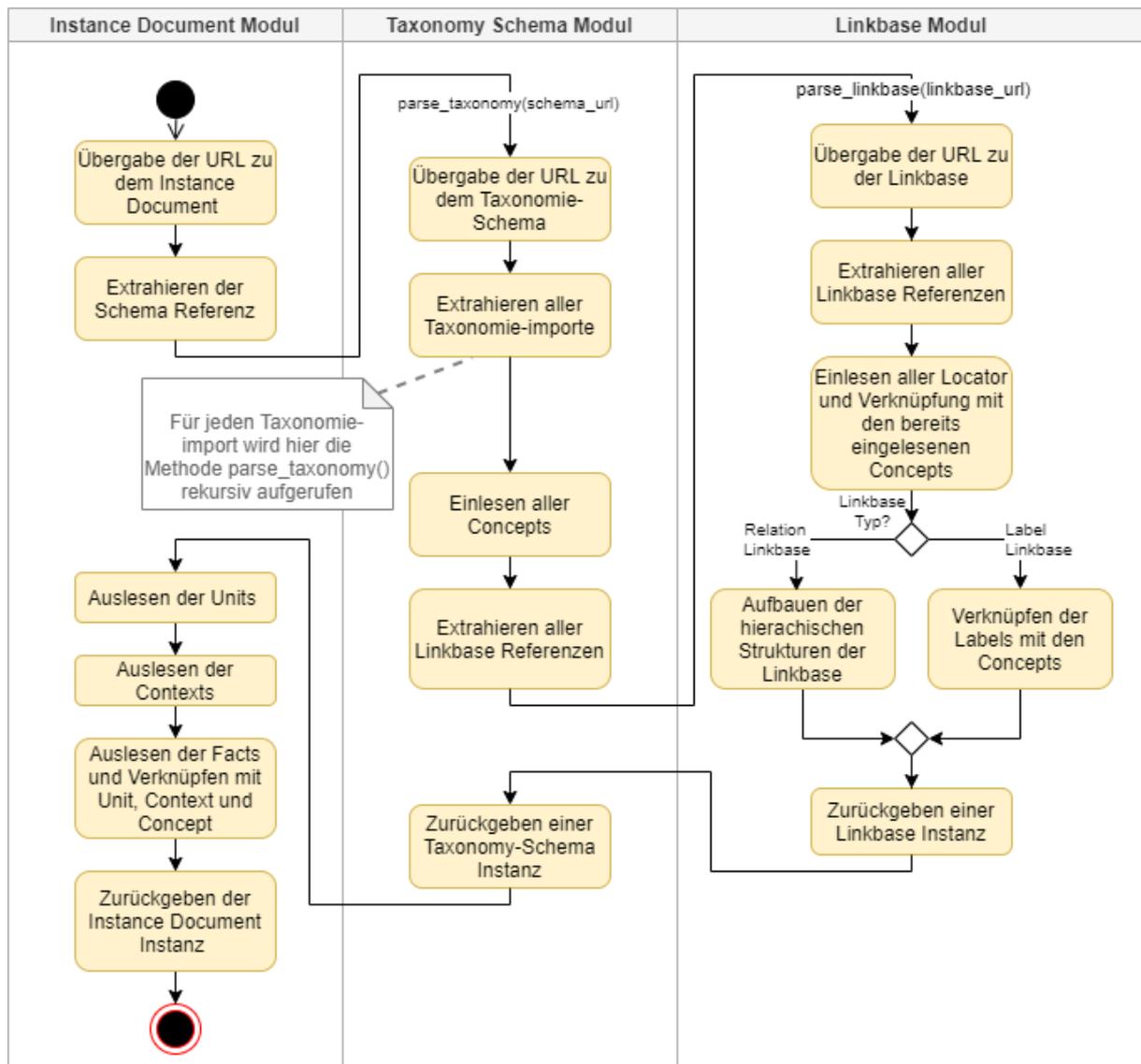


Abbildung 11: Aktivitätsdiagramm des XBRL-Parsers<sup>93</sup>

Im ersten Schritt wird eine URL zu dem XBRL-Bericht an das *Instance Document* Modul übergeben. Der Bericht wird dann mit dem XML-Parser eingelesen, um die Schema Referenz zu erhalten. Diese wird anschließend an das Taxonomie-Schema Modul übergeben. Das Taxonomie-Schema wird auch hier zuerst mit dem XML-Parser eingelesen. An dieser Stelle wird einfachheitshalber davon ausgegangen, dass keine weiteren Taxonomien importiert werden. Daher kann direkt mit dem Einlesen der Konzepte weitergemacht werden.

<sup>93</sup> Eigene Darstellung

Nachdem die Konzepte eingelesen sind, können die *Linkbases* eingelesen werden. Hier wird wieder nur die URL an das *Linkbase* Modul übergeben. Dort werden alle *Locators* eingelesen und mit den vorher eingelesenen Konzepten verknüpft. Danach wird unterschieden, ob es sich um eine *Label Linkbase* oder eine *Relation Linkbase* (*Calculation*-, *Presentation*- oder *Definition Linkbase*) handelt. Bei einer *Label Linkbase* werden jeweils alle Labels gesucht, welche von einem *Locator* referenziert werden. Diese können dann mit dem Konzept verknüpft werden. Bei den *Relation Linkbases* werden die *Locator-Arc* Beziehungen hierarchisch im Objekt abgespeichert. Wenn die *Linkbase* eingelesen ist, wird von dem Modul eine *Linkbase* Objektinstanz zurückgegeben, welche im *Taxonomie-Schema* Objekt abgespeichert wird.

Nach erfolgreichem Einlesen der *Taxonomie* kann mit der Verarbeitung des *Instance Documents* weitergemacht werden. Hier muss mit dem Einlesen der zeitlichen *XBRL-Kontexte* und der *XBRL-Einheiten* begonnen werden. Im letzten Schritt können dann die einzelnen *Facts* des *Instance Documents* mit den Konzepten der vorher eingelesenen *Taxonomien*, sowie mit den *Einheiten* und *Kontexten* verknüpft werden.

Im Folgenden soll das Einlesen der *Taxonomien* nochmal genauer behandelt werden. *Abbildung 12* zeigt die verschiedenen Schritte für eine *Taxonomie-Erweiterung*, welche zwei weitere *Taxonomien* importiert, auf. Diese Struktur findet man sehr häufig in einer offenen *Berichtserstattung*. So könnte beispielsweise die *Taxonomie 1* eine *US-GAAP Taxonomie* sein und die *Taxonomie 2* eine *Stammdaten Taxonomie*, mit welcher allgemeine Informationen zum berichtenden Unternehmen gekennzeichnet werden können.

Wichtig beim Einlesen der *Taxonomien* ist, dass die unterliegenden *Taxonomien* immer zuerst komplett eingelesen werden, da die *Taxonomie-Erweiterung* eventuell Strukturen oder Ressourcen der *Taxonomie 1* und *2* überschreibt. So könnte z.B. eine *Fluggesellschaft* das Label „Zu veräußernde langfristige Vermögenswerte“ der *Taxonomie 1* in der *Taxonomie-Erweiterung* mit dem Label „Verminderung der Flotte“ überschrieben. Damit dieses Label in der *Taxonomie-Erweiterung* überschrieben werden kann (*Schritt 7*), muss es also vorher eingelesen werden (*Schritt 4*).

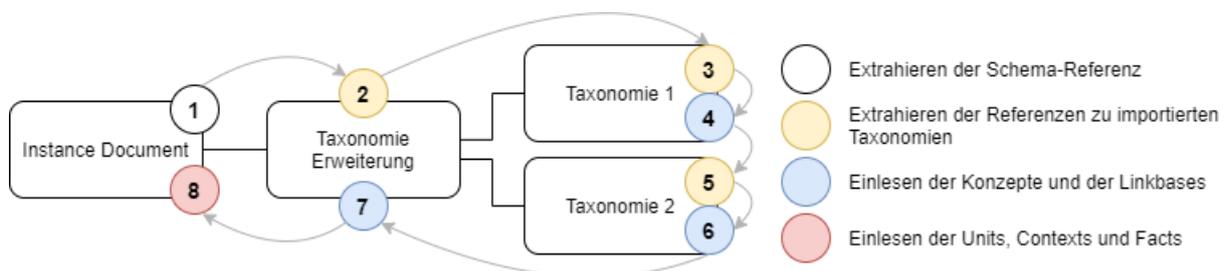


Abbildung 12: Ablauf beim Einlesen der Taxonomie-Importe<sup>94</sup>

<sup>94</sup> Eigene Darstellung

Um diese Reihenfolge des Einlesens zu erreichen, wurde auf eine rekursive Implementierung der „`parse_taxonomy()`“ Methode zurückgegriffen die im Folgenden erklärt wird.

Bevor das Taxonomie-Schema eingelesen werden kann, muss diese erst heruntergeladen werden. Dies geschieht über den Aufruf „`cache_file()`“ in Zeile 4 des folgenden Codebeispiels 15. Die Klasse „`CacheHelper`“ setzt die im vorherigen Kapitel angesprochene Caching-Strategie um und gibt einen lokalen Dateipfad zurück, von dem die Datei ohne weitere Netzwerkabfrage direkt eingelesen werden kann. Nun kann die Datei mit dem XML-Parser eingelesen werden. Als XML-Parser wurde hier `ElementTree` verwendet. Nach dem Einlesen der Datei (Zeile 7) werden zuerst alle Taxonomie-Importe gesucht (Zeile 11) und dann einzeln durchgegangen. Pro Taxonomie-Import wird die „`parse_taxonomy()`“ Methode rekursiv aufgerufen (Zeile 18), sodass am Rekursionsende alle unterliegenden Taxonomien vollständig eingelesen sind.

Der Aufruf der Methode „`resolve_uri()`“ der Klasse „`URIHelper`“ in Zeile 17 ist nötig, da manche Taxonomie-Schemas über eine relative URL (z.B. „`../ifrs-core.xsd`“) anstatt über eine absolute URL (z.B. „`http://example.com/ifrs-code.xds`“) verlinkt werden. In diesem Fall muss über die URL des aktuellen Schemas erst die absolute URL herausgefunden werden, damit diese später vom `CacheHelper` heruntergeladen werden kann.

Die komplette Funktion „`parse_taxonomy()`“ wird mit einem LRU-Cache versehen (Zeile 1). Da bei der Verarbeitung von vielen XBRL-Berichten bestimmte Taxonomien immer wieder benötigt werden (z.B. die US-GAAP Taxonomie), können diese nach einmaligem Einlesen direkt aus dem Cache entnommen werden. Dadurch müssen die eigentlichen XML-Dateien der Taxonomie nicht mehr eingelesen werden was sich positiv auf die Laufzeit auswirkt.

```

1  @lru_cache(maxsize=60)
2  def parse_taxonomy(schema_url: str) -> TaxonomySchema:
3      try:
4          schema_path: str = CacheHelper.cache_file(schema_url)
5      except FileNotFoundError:
6          raise TaxonomyNotFound(f"Couldn't find schema from {schema_url}")
7      root: ET.Element = ET.parse(schema_path).getroot()
8      target_ns = root.attrib['targetNamespace']
9      taxonomy: TaxonomySchema = TaxonomySchema(schema_url, target_ns)
10
11     import_elements: [ET.Element] = root.findall('xsd:import', NS)
12     for import_element in import_elements:
13         import_url: str = import_element.attrib['schemaLocation']
14         # The url to the imported schema can also be relative !
15         if not import_url.startswith('http'):
16             # get the absolute URL
17             import_url = URIHelper.resolve_uri(schema_url, import_url)
18         taxonomy.imports.append(parse_taxonomy(import_url))
19     # start parsing all concepts
20     [...]
```

Codebeispiel 15: *Einlesen der importierten Taxonomien*

Nach dem Einlesen aller unterliegenden Taxonomie-Schemas werden die Konzepte und deren Attribute eingelesen. Dies kann mit einem einfachen Aufruf der Methode „`findall()`“ des XML-Parsers geschehen, da die Konzepte alle in derselben Ebene definiert sind und keine hierarchische Struktur aufweisen. Gespeichert werden die Konzepte in einem Python Dictionary innerhalb der Taxonomie-Objektinstanz mit deren ID als Schlüssel. Dadurch wird ein Zugriff auf eine Konzept Objektinstanz in konstanter Laufzeit möglich, solange man die ID des Konzeptes hat.

Im Anschluss werden die Referenzen auf die Linkbases herausgesucht und dem Linkbase Modul übergeben. Dies sieht von der Implementierung im Programmcode ähnlich aus wie die Zeilen 11 bis 18 des Codebeispiels 15, mit dem Unterschied, dass hier XML Elemente mit dem Namen „`link:linkbaseRef`“ gesucht werden und die URL's dieser referenzierten Linkbases dem Linkbase Module mit dem Aufruf „`parse_linkbase()`“ übergeben werden.

Beim Einlesen der Linkbases unterscheidet der Parser zwischen den Relation Linkbases und den Label Linkbases. Reference Linkbases werden nicht betrachtet. Wie schon im Kapitel „7.7.2.1 Relation Linkbases“ besprochen, bauen Relation Linkbases hierarchische Strukturen zwischen Konzepten auf, während Label Linkbases nur ein Konzept mit einer oder mehreren Ressourcen verknüpfen. Diese Strukturen bzw. Verknüpfungen mit Ressourcen werden ebenfalls in einer Instanz der Klasse „`Linkbase`“ abgespeichert und an das Taxonomie-Schema Modul zurückgegeben. Abbildung 13 zeigt einen Ausschnitt einer eingelesenen *Calculation Linkbase* zur Laufzeit. Hier werden dem Konzept „`AssetsCurrent`“ die Konzepte in dem Array „`children`“ untergeordnet. Die Objektinstanzen in dem Array „`children`“ sind dabei von der Klasse „`CalculationArc`“. Diese entspricht dem XLink Arc-Element aus der XML-Datei und hält weitere Informationen über die rechnerische Beziehung. Hier findet man beispielsweise das vorher angesprochene Attribut „`weight`“, mit welchem eine Addition („`weight = 1`“) oder eine Subtraktion („`weight = -1`“) gekennzeichnet werden kann.

```

{.-} 0 = {Locator} loc_us-gaap_AssetsCurrent_6be4a026-2325-49eb-a729-19fb3e6dc9b2 with 6 children
  concept_id = {str} 'us-gaap_AssetsCurrent'
  href = {str} 'http://xbrl.fasb.org/us-gaap/2020/elts/us-gaap-2020-01-31.xsd#us-gaap_AssetsCurrent'
  name = {str} 'loc_us-gaap_AssetsCurrent_6be4a026-2325-49eb-a729-19fb3e6dc9b2'
  children = {list: 6} [<__main__.CalculationArc object at 0x000001BFE9BFF5E0>, <__main__.CalculationArc object at 0x000001BFE9BFF5E0>]
  _len_ = {int} 6
  
```

Abbildung 13: Hierarchische Struktur einer eingelesenen *Calculation Linkbase*<sup>95</sup>

<sup>95</sup> Eigene Darstellung

Nachdem alle Taxonomien mit ihren Linkbases eingelesen wurden, kann mit dem eigentlichen Auslesen des *Instance Documents* begonnen werden. Hier werden zuerst alle Kontexte und Einheiten eingelesen und wieder in Dictionaries mit ihrer ID als Schlüssel abgespeichert.

```
1 <us-gaap:Revenues contextRef="FD2020Q1TD" decimals="-6" unitRef="usd">
2   78351000000</us-gaap:Revenues>
```

Codebeispiel 16: Deklaration eines Facts

Codebeispiel 16 zeigt nochmal die Deklaration eines möglichen Facts. Anhand diesem sollen die letzten Schritte des eigentlichen Einlesens der Daten erklärt werden.

1. Über den Präfix „us-gaap“ und dem dazugehörigen Namespace wird die Taxonomie aufgefunden in welcher das Konzept definiert worden ist. Dies wird durch einen Aufruf der Methode „get\_taxonomy()“ (Codebeispiel 17) gemacht. Hier werden alle vorher importierten Taxonomien rekursiv durchsucht, bis die Taxonomie mit dem gesuchten Namespace gefunden worden ist.

```
1 def get_taxonomy(self, namespace: str) -> TaxonomySchema or None:
2     if self.namespace == namespace:
3         return self
4
5     for imported_tax in self.imports:
6         result = imported_tax.get_taxonomy(namespace)
7         if result is not None:
8             return result
9     return None
```

Codebeispiel 17: Funktion zur rekursiven Suche einer Taxonomie

2. Nachdem die gesuchte Taxonomie zurückgegeben wurde, kann über die ID das Konzept aus der Taxonomie abgerufen werden (Zeile 4). Ebenso kann der XBRL-Kontext aus dem vorher extrahiertem Dictionary herausgelesen werden. Dies wird über den Wert des Attributes „contextRef“ des Facts gemacht (Zeile 5).

```
1 tax = taxonomy.get_taxonomy(taxonomy_ns)
2 if tax is None: raise TaxonomyNotFound(taxonomy_ns)
3
4 concept: Concept = tax.concepts[concept_id]
5 context: AbstractContext = context_dir[elm.attrib['contextRef']]
```

Codebeispiel 18: Abrufen der für den Fact benötigten Informationen

3. Das Einlesen der Einheit funktioniert nach demselben Schema. Allerdings muss man hier beachten, dass nicht jeder Fact mit einer Einheit verbunden ist. So gibt es beispielsweise XBRL-Facts, welche ein Datum als Wert halten und deswegen keine Einheit besitzen.

### 8.5.4 Zwischenspeichern der Facts

Nachdem die XBRL-Berichte durch den Parser in die Objektinstanzen überführt worden sind, kann mit diesen weitergearbeitet werden ohne weitere Dateien einlesen zu müssen. Die Objektinstanzen können beispielsweise über Schnittstellen in andere Systeme eingeführt werden oder in einer Datenbank abgespeichert werden. In dem Anwendungsbeispiel wurde eine Teilmenge der Facts aus den *Instance Documents* in eine MySQL Datenbank überführt. Hier können dann weitere Abfragen getätigt werden.

Abbildung 14 zeigt den Aufbau der benutzten Tabellen. Die extrahierten XBRL-Facts werden in der Tabelle „facts“ abgespeichert. Hierbei wird die Struktur stark vereinfacht. Die Tabelle speichert nur Facts die das Unternehmen, welches den Bericht erstellt hat, direkt betreffen. Facts von Unternehmensbereichen oder Subunternehmen werden ignoriert. Zudem wird der zeitliche Kontext des Facts stark vereinfacht. Statt ein Start- und Enddatum abzuspeichern, wird nur das Enddatum abgespeichert. Die Zeitspanne zwischen Start- und Enddatum wird dafür in Quartale umgerechnet. Dadurch werden Abfragen sehr viel einfacher, da es nun möglich ist, jährliche Facts über „quarters=4“ abzurufen, anstatt bei jeder Abfrage die Zeitspanne in Jahre umrechnen zu müssen. Facts welche sich auf einen bestimmten Zeitpunkt beziehen (z.B. Bilanzstichtag), werden mit „quarters=0“ abgespeichert.

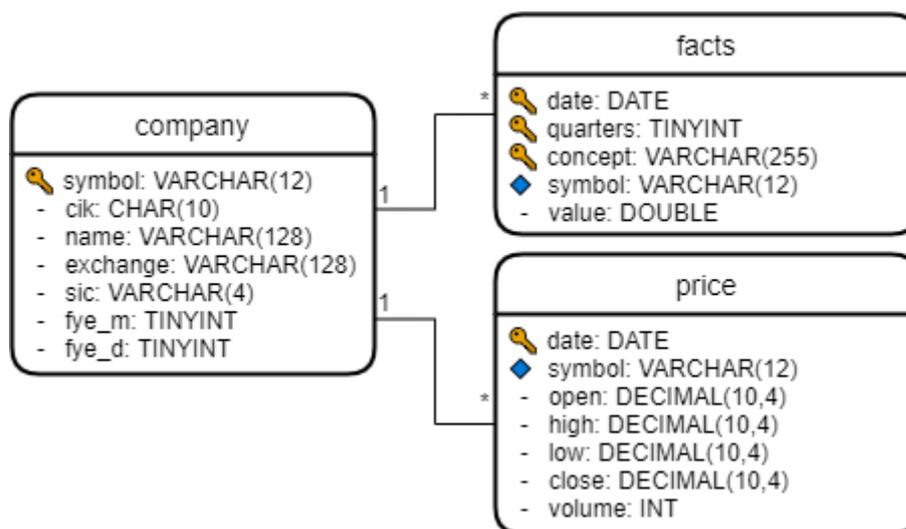


Abbildung 14: Struktur der Tabellen zum Zwischenspeichern<sup>96</sup>

Eine weitere wichtige Spalte der **Tabelle „facts“** aus Abbildung 14 ist die Spalte „concept“. Hier wird die ID des Konzeptes abgespeichert, mit dem der Fact im *Instance Document* gekennzeichnet worden ist. Die Spalte „symbol“ enthält das Tickersymbol unter welcher das Unternehmen an der Börse gehandelt wird.

In der **Tabelle „company“** ist vor allem die Spalte „sic“ wichtig. Diese existiert auch in der vorher erklärten Tabelle „sub\_index\_items“ des EDGAR RSS Disclosure Feeds. Dort heißt die Spalte „assigned\_sic“. SIC (Standard Industrial Classification) ist das Klassifikationsschema mit welchem die Unternehmen in verschiedene Industrien eingeordnet werden können.<sup>97</sup>

<sup>96</sup> Eigene Darstellung

<sup>97</sup> Eine Übersicht der SIC Codes findet man unter <https://siccode.com/sic-code-lookup-directory>

Mit dem SIC-Code werden im nächsten Kapitel die Unternehmen verschiedenen Industrien zugeordnet. Die für die Tabelle „company“ benötigten Informationen werden aus dem vorher behandelten monatlichen RSS Feed sowie aus dem letzten XBRL-Bericht entnommen.

In der **Tabelle „price“** werden die Informationen über den Preis der Aktie des Unternehmens gespeichert. Diese Aktienkursdaten haben nichts mit den extrahierten XBRL-Daten zu tun, werden aber für die Beispielanalyse benötigt. Die Daten hierfür wurden über die API des Anbieters „IEX Cloud“ bezogen.

### 8.5.5 Beispielhafte Analyse

Mit den extrahierten Finanzinformationen können nun eine Vielzahl an Analysen durchgeführt werden. Im Rahmen dieser Arbeit soll nur eine Beispielanalyse erstellt werden, welche nur einen sehr kleinen Anwendungsfall abdeckt.

In dieser Beispielsanalyse werden durchschnittliche Kurs-Gewinn-Verhältnisse (KGV) von unterschiedlichen Sektoren verglichen. Das KGV ist die von Analysten am häufigsten verwendete Kennzahl beim Analysieren eines Unternehmens<sup>98</sup> und misst die Bewertung eines Unternehmens an der Börse. Sie lässt sich berechnen, indem man den Aktienkurs des Unternehmens durch den Gewinn pro Aktie dividiert.

$$\text{Kurs-Gewinn Verhältnis (KGV)} = \frac{\text{Kurs der Aktie (am Tag X)}}{\text{Gewinn je Aktie (für das Jahr Y)}}$$

Abbildung 15: Formel des KGV<sup>99</sup>

Da in der Beispielanalyse das KGV jeweils zum Ende von 2019 und 2020 ermittelt werden soll, benötigt man den Gewinn pro Aktie über vier Quartale sowie den letzten Aktienkurs des Jahres.

Diese Informationen können nun aus den im vorherigen Kapitel vorgestellten Tabellen herausgelesen werden. Das folgende Codebeispiel 19 zeigt den SQL-Befehl mit welchen die benötigten Daten für alle Unternehmen abgefragt werden können. Hier werden die „financial“, die „price“ und die „company“ Tabelle über einen INNER JOIN verbunden (Zeile 3 - 5). Dadurch werden erstmal nur Einträge zurückkommen, bei welchen XBRL-Facts, der Aktienkurs und die generellen Unternehmensinformationen vorliegen. Diese werden über die folgenden Bedingungen weiter eingegrenzt. Zuerst werden alle Facts selektiert, welche mit dem Konzept „us-gaap\_EarningsPerShare“ (Gewinn pro Aktie nach US-GAAP) gekennzeichnet werden (Zeile 6). Außerdem wird über die Bedingung „quarters=4“ sichergestellt, dass nur Facts die sich auf das gesamte Geschäftsjahr (4 Quartale) beziehen zurückgegeben werden. Die folgenden Bedingungen mit den Unterabfragen gehen sicher, dass jeweils der letzte verfügbare Datenpunkt für das Jahr zurückgegeben wird (Zeile 9 - 21).

<sup>98</sup> Vgl. Heese, Viktor: Aktienbewertung Mit Kennzahlen: Kurschancen und -risiken fundiert beurteilen, Wiesbaden, Deutschland: Gabler Verlag, 2011, S. 33.

<sup>99</sup> Heese, 2011, S. 43.

```

1 SELECT comp.symbol, comp.sic, fin.date as eps_date, fin.value as
2   eps, pri.date as price_date, pri.close as stock_price
3 FROM financial fin
4 INNER JOIN price pri ON(fin.symbol = pri.symbol)
5 INNER JOIN company comp ON(fin.symbol = comp.symbol)
6 WHERE concept = 'us-gaap_EarningsPerShareBasic'
7 AND quarters = 4
8 -- Get the most current EPS for the year 2020
9 AND (fin.symbol, fin.date) IN (
10  SELECT symbol, MAX(date)
11  FROM financial
12  WHERE concept = 'us-gaap_EarningsPerShareBasic'
13  AND date BETWEEN '2020-09-01' AND '2020-12-31'
14  AND quarters = 4 GROUP BY symbol)
15 -- Get the most current stock price for the year 2020
16 AND (pri.symbol, pri.date) IN(
17  SELECT symbol, MAX(date)
18  FROM price
19  WHERE date BETWEEN '2020-12-01' AND '2020-12-31'
20  GROUP BY symbol)
21 AND sic IS NOT NULL;

```

*Codebeispiel 19: SQL-Befehl zum Abrufen von Gewinn pro Aktien und Aktienkurs*

Im Anwendungsbeispiel wurde die SQL-Abfrage mithilfe von Pandas eingelesen. Pandas ist eine populäre Python Bibliothek, welche eine große Anzahl an nützlichen Funktionen für die Datenanalyse und Datenverarbeitung bietet. Mit einem Pandas DataFrame können Daten in einer tabellarischen Struktur abgespeichert werden und leicht verarbeitet werden. Im Codebeispiel 20 wurde das DataFrame „df“ genannt. Es enthält die Ergebnisse der vorher gezeigten SQL-Abfrage.

```

1 df['kgv'] = df['stock_price'] / df['eps']
2 [...]
3 grouped_df = df.groupby('industry')['kgv'].apply(list)
4 [...]
5 plot.boxplot(grouped_df, labels=labels, showfliers=False, vert=False)

```

*Codebeispiel 20: Berechnung und Visualisierung der KGVs*

Zeile 1 zeigt die Berechnung des KGV. Hier wird der letzte Aktienkurs durch den jährlichen Gewinn pro Aktie (im Englischen „Earnings per Share - **eps**“) dividiert, wodurch man das Kurs-Gewinn Verhältnis erhält. Danach folgen einige Zeilen Code, die für das Bereinigen der Daten und das Entfernen von extremen Ausreißern notwendig sind. Diese wurden hier einfachheitshalber ausgelassen. Zeile 3 zeigt eine Gruppierung des Kurs-Gewinn-Verhältnisses nach Industrie. Der zurückgegebene Wert dieses Aufrufes hat nun für jede Industrie ein Array, welches die KGVs aller Unternehmen der jeweiligen Industrie enthält. Anschließend werden die nach Industrie gruppierten KGV's über die Methode „boxplot“ der Python Bibliothek „matplotlib“ visualisiert.

Der daraus resultierende Box-Plot wird in Abbildung 16 gezeigt. Die gestrichelten Linien zwischen den Sektoren wurden zur besseren Übersicht nachträglich eingefügt.

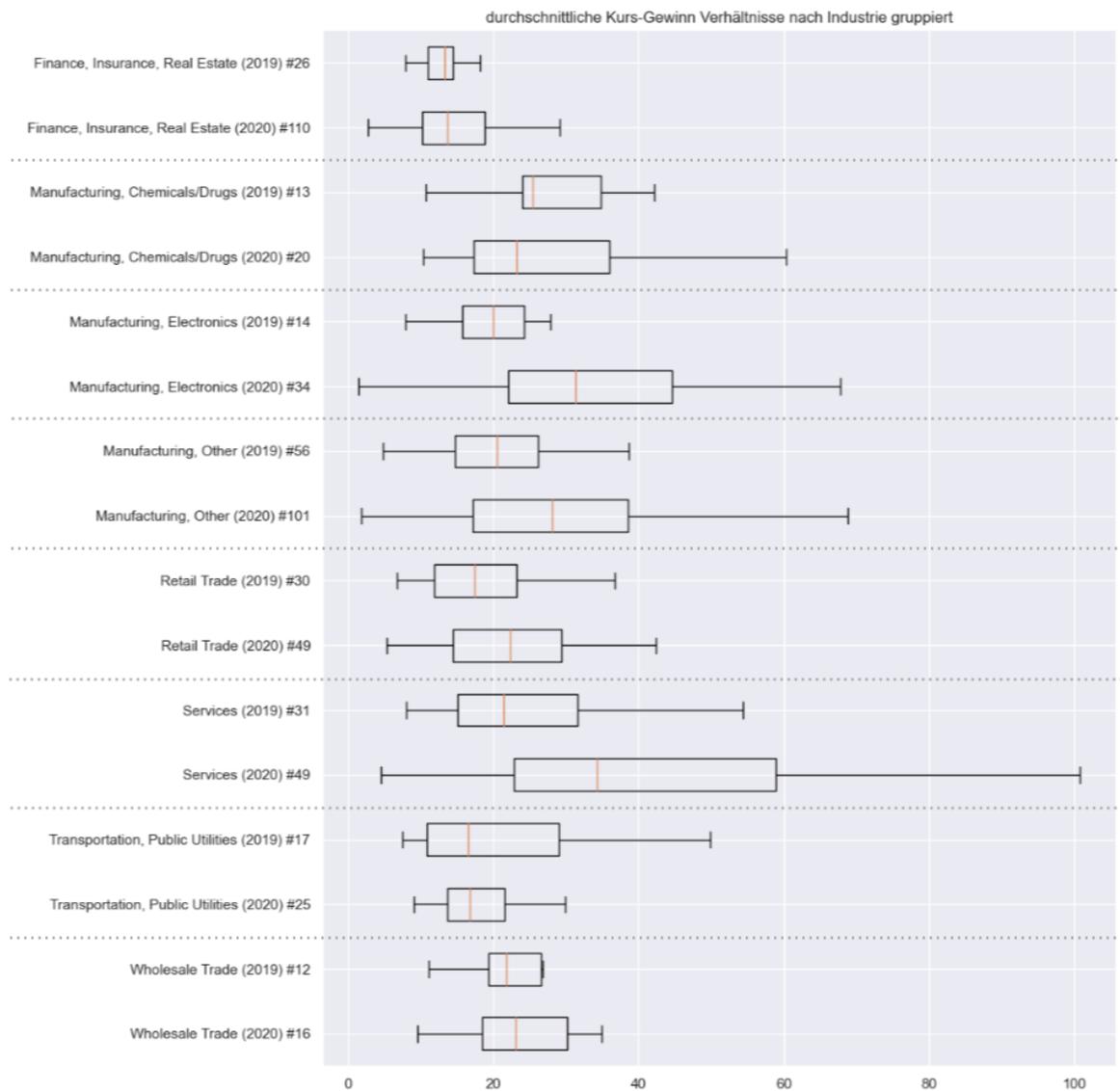


Abbildung 16: Ergebnis der KGV-Analyse<sup>100</sup>

Anhand dieses Diagrammes kann man ableiten, wie bestimmte Sektoren am Aktienmarkt bewertet sind. Auffällig ist beispielsweise, dass der Dienstleistungsbereich im Vorjahresvergleich stark an Bewertung gewonnen hat.

Das hier gezeigte Diagramm ist aber kritisch zu betrachten, da nicht alle Unternehmen der einzelnen Sektoren mit einbezogen werden konnten. Die Zahl hinter dem Industrienamen zeigt, wie viele Unternehmen in dem jeweiligen Sektor betrachtet werden konnten.

Die geringe Anzahl an Unternehmen pro Sektor ist aber nicht auf fehlende Fundamentaldaten zurückzuführen (die Unternehmen sind ja gesetzlich dazu verpflichtet ihre Finanzberichte bei der SEC einzureichen), sondern wurde durch fehlende Aktienkurse bedingt. Diese fehlenden Aktienkurse konnten nicht über die IEX Cloud API abgerufen werden, da monatliche Limit für API-Anfragen erreicht worden ist. Dadurch war die Berechnung des KGVs für einen großen Teil der Firmen nicht möglich.

<sup>100</sup> Eigene Darstellung

## 9 Kritische Betrachtung

Wenn auch die Adaption der XBRL-Technologie für die Datenübertragung von finanziellen Informationen erhebliche Prozessverbesserungen ermöglicht, gibt es einige Kritikpunkte, die man beachten sollte. Im Folgenden werden diese weiter erläutert.

Die Umstellung des Meldewesens auf den XBRL Standard hilft nicht bei der Bekämpfung von Manipulation oder dem Verhindern von Falschaussagen gegenüber den Behörden. Auch mit einer XBRL-basierten finanziellen Berichtserstattung wird es nach wie vor möglich sein, Informationen aktiv zu verfälschen.<sup>101</sup> Mit XBRL ist lediglich eine Prüfung des *Instance Documentes* gegen eine Taxonomie durchführbar, womit der Bericht nur auf rechnerische und strukturelle Richtigkeit überprüft werden kann.

Die Qualität eines XBRL-Berichtes wird stark durch die Taxonomie bedingt. Wenn wichtige Konzepte nicht in der Basis-Taxonomie enthalten sind, können sie nicht im Bericht gekennzeichnet werden bzw. müssen in einer eigenen Erweiterungs-Taxonomie deklariert werden. Wenn allerdings jedes Unternehmen ein neues Konzept erstellt, können diese nicht einfach unternehmensübergreifend verglichen werden, obwohl sie denselben Sachverhalt darstellen. Dieses Problem tritt z.B. in der IFRS Taxonomie auf. Hier werden keine Pro-Forma-Kennzahlen wie EBIT definiert, wodurch man aufgrund von eventuell unterschiedlichen Berechnungsmethoden diese Kennzahlen nicht unternehmensübergreifend vergleichen kann.<sup>102</sup>

Investoren profitieren vor allem sekundär von einer Umstellung auf XBRL. Beispielsweise durch eine Verbesserung der Verbreitungsgeschwindigkeit der Unternehmensdaten oder durch verbesserte Softwarefeatures. Das eigenständige Auswerten von XBRL-Berichten, wie es im Anwendungsbeispiel durchgeführt wurde, ist jedoch mit sehr viel Aufwand verbunden. Dies ist vor allem durch die sehr hohen Speicherplatzanforderungen bedingt, welche für die XBRL Berichte benötigt werden. Mit einer durchschnittlichen Wortanzahl eines Jahresberichtes von 42000 Wörtern (im Jahr 2017)<sup>103</sup> und weiteren dazukommenden Taxonomien und Linkbasen kommt hierbei eine beachtliche Dateigröße pro Bericht zusammen. Für das Anwendungsbeispiel in Kapitel 8 wurden alle Quartals- und Jahresberichte, welche im Zeitraum von Anfang 2018 bis Ende 2020 bei der SEC eingereicht wurden, heruntergeladen. Hierfür mussten 221843 Dateien heruntergeladen werden, welche zusammen 114 GB Speicherplatz benötigt hatten. Darunter fallen auch jegliche von den Berichten importierte Taxonomien, welche im Vergleich mit den XBRL-Dateien der einzelnen Finanzberichte jedoch nur einen kleinen Anteil ausmachen. Dieser Aufwand ist für einen einzelnen Investor, welcher unternehmensübergreifende Analysen durchführen will, viel zu hoch. Er könnte allerdings

---

<sup>101</sup> Vgl. Pfister/Venet, 2009, S. 436.

<sup>102</sup> Vgl. Kovarova-Simecek, Monika/Tassilo Pellegrini/Tatjana Aubram: CARF Luzern 2018: Controlling.Accounting.Risiko.Financen., in: CARF Luzern 2018 Konferenzband, 2018, [online] <https://financial-communications.fhstp.ac.at/wp-content/uploads/2018/09/xbrl-im-digitalen-reporting-entwicklungsstand-in-europa-und-in-österreich.pdf>, S. 14.

<sup>103</sup> Vgl. Lesmy, Danny/Lev Muchnik/Yevgeny Mugeran: Doyoureadme? Temporal Trends in the Language Complexity of Financial Reporting, in: SSRN Electronic Journal, 2019, doi: 10.2139/ssrn.3469073, S. 4.

sekundär durch Software, die beispielsweise den Vergleich von zwei Finanzberichten ermöglicht, profitieren.

Ein weiterer Kritikpunkt ist die geringe Verfügbarkeit von OpenSource Projekten im XBRL-Bereich. Zudem lohnt sich eine Eigenentwicklung eines XBRL-Prozessors wirtschaftlich oft nicht, da dies mit sehr viel Entwicklungsaufwand verbunden ist. Daher ist es in vielen Fällen sinnvoll auf kommerzielle Software zu setzen.

Zuletzt müssen auch noch die unternehmensinternen Aufwände bei der Einführung des XBRL-Standards angesprochen werden. Am Beispiel der Einführung des ESEF sieht man aktuell wie schwierig die Umstellung des Meldewesens auf XBRL sein kann. Unternehmen müssen teilweise ihre gesamten Buchhaltungssysteme überarbeiten, um ein automatisches Kennzeichnen der einzelnen Positionen mit XBRL Konzepten zu ermöglichen. Zudem werden häufige Änderungen der ESEF-Taxonomie bei den einzelnen Unternehmen einen erheblichen Mehraufwand erzeugen.<sup>104</sup>

## 10 Fazit

Der XBRL-Standard hat sich als Übertragungsstandard von Finanzinformationen etabliert. Mit ihm lassen sich viele Szenarien im Meldewesen und darüber hinaus abdecken. Er ermöglicht unter anderem das maschinelle Auslesen von Finanzberichten, wodurch manuelle Datenübertragungsprozesse standardisiert und automatisiert werden können. Die Einführung des XBRL-Standards kann somit zu einer erheblichen Steigerung der Effizienz dieser Prozesse führen. Ein großer Vorteil des XBRL-Standards ist die Modularität und Erweiterbarkeit. So können einzelne Taxonomie Module für unterschiedliche Industrien erstellt werden, um die Meldeanforderungen der jeweiligen Industrie genau abzubilden. Dies ist vor allem in der Finanzbranche wichtig, da hier meist strengere Anforderungen an das Berichtswesen gelten. Falls die in der Taxonomie enthaltenen Konzepte immer noch nicht ausreichen, wird in einer offenen Berichtserstattung dem Unternehmen ermöglicht, durch Taxonomie-Erweiterungen unternehmensspezifische Konzepte zu erstellen und diese zu verwenden. Beachten sollte man aber auch den großen Aufwand der einzelnen Unternehmen bei der Einführung eines auf XBRL-basierenden Meldewesens.

Im Anwendungsbeispiel wurde aufgezeigt, wie aufwendig das Aufstellen von unternehmensweiten Analysen auf Basis von XBRL-Dokumenten sein kann. Hier stellen vor allem die immense Menge und Größe der benötigten Dateien sowie fehlende Open Source Lösungen große Hürden auf.

---

<sup>104</sup> Vgl. Müller, Stefan/Patrick Saile: ESEF: Praktische Implikationen – die Zeit läuft, in: Haufe.de News und Fachwissen, 30.09.2020, [online] [https://www.haufe.de/finance/jahresabschluss-bilanzierung/digitalisierung-von-finanzberichten-esef/praktische-implikationen-fruehzeitige-vorbereitung-ist-ratsam\\_188\\_455366.html](https://www.haufe.de/finance/jahresabschluss-bilanzierung/digitalisierung-von-finanzberichten-esef/praktische-implikationen-fruehzeitige-vorbereitung-ist-ratsam_188_455366.html) [12.01.2021].

## 11 Literaturverzeichnis

- Althoff, Frank/Andreas Arnold/Arne Jansen/Tobias Polka/Frank Wetzel: Die neue E-Bilanz, 1. Aufl., Freiburg, Deutschland: Haufe-Lexware GmbH, 2011.
- Debreceny, Roger/Carsten Felden/Bartosz Ochocki/Maciej Piechocki/Michal Piechocki: XBRL for Interactive Data: Engineering the Information Value Chain, 2009. Aufl., Heidelberg, Deutschland: Springer, Berlin, Heidelberg, 2009.
- Debreceny, Roger/Carsten Felden/Maciej Piechocki: New Dimensions of Business Reporting and XBRL, 1. Aufl., Wiesbaden, Deutschland: Deutscher Universitätsverlag, 2007.
- Delivorias, Angelos/Europäischen Parlament: Prospectuses for investors – Simplifying equity-raising during the pandemic, in: Europäisches Parlament, 19.11.2020, [online] [https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/659350/EPRS\\_BRI\(2020\)659350\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/659350/EPRS_BRI(2020)659350_EN.pdf) [10.01.2021].
- Enachi, Mihaela/Ioan I. Andone: The Progress of XBRL in Europe – Projects, Users and Prospects, in: Procedia Economics and Finance, Jg. 2015, Nr. 20, 2015, doi: 10.1016/s2212-5671(15)00064-7, S. 185–192.
- Europäische Kommission: Bericht der Kommission an das europäische Parlament und den Rat, in: EUR-LEX, 27.11.2012, [online] <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2012:0699:FIN:DE:PDF> [10.12.2020].
- Europäisches Parlament: A simplified business environment for companies - P6\_TA(2008)0220, in: Website des Europäischen Parlamentes, 21.05.2008, [online] <https://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+TA+P6-TA-2008-0220+0+DOC+XML+V0//EN> [01.12.2020].
- European Securities and Markets Authority: Final Report on the RTS on the European Single Electronic Format, in: ESMA Website, 18.12.2017, [online] [https://www.esma.europa.eu/sites/default/files/library/esma32-60-204\\_final\\_report\\_on\\_rts\\_on\\_esef.pdf](https://www.esma.europa.eu/sites/default/files/library/esma32-60-204_final_report_on_rts_on_esef.pdf) [26.11.2020].
- Fischer, Herm/Mark V Systems Limited/XBRLSpy Research Inc./Diane Mueller: Enabling Comparability and Data Mining with the Arelle® Open Source Unified Model, in: Arelle, 2011, [online] <https://arelle.org/arelle/download/1014/> [02.01.2021].
- Flickinger, Norbert: XBRL in der betrieblichen Praxis: Der Standard für Unternehmensreporting und E-Bilanz, Berlin, Deutschland: Erich Schmidt Verlag, 2013.
- Gerdes, John: EDGAR-Analyzer: automating the analysis of corporate data contained in the SEC's EDGAR database, in: Decision Support Systems, Jg. 2003, Nr. 35, 2003, doi: 10.1016/s0167-9236(02)00096-9, S. 7–29.

- Goodhand, Mark/Charles Hoffman/Brad Homer/Josif MacDonald/Geoff Shuetrim/Hugh Wallis: XBRL International, in: Financial Reporting Taxonomies Architecture 1.0, 29.01.2005, [online] <http://www.xbrl.org/technical/guidance/old/frta-cr5-2005-01-29.pdf> [22.12.2020].
- Heese, Viktor: Aktienbewertung Mit Kennzahlen: Kurschancen und -risiken fundiert beurteilen, Wiesbaden, Deutschland: Gabler Verlag, 2011.
- Kovarova-Simecek, Monika/Tassilo Pellegrini/Tatjana Aubram: CARF Luzern 2018: Controlling.Accounting.Risiko.Financen., in: CARF Luzern 2018 Konferenzband, 2018a, [online] <https://financial-communications.fhstp.ac.at/wp-content/uploads/2018/09/xbrl-im-digitalen-reporting-entwicklungsstand-in-europa-und-in-osterreich.pdf>, S. 1–16.
- Kovarova-Simecek, Monika/Tassilo Pellegrini/Tatjana Aubram: XBRL im digitalen Reporting - Entwicklungsstand in Europa und in Österreich, in: Linard Nadig/Ulrich Egle (Hrsg.), CARF Luzern 2018: Controlling.Accounting.Risiko.Financen., 2018b, [online] [https://www.wiwi.uni-siegen.de/banken/dokumente/konferenzband\\_carf\\_luzern\\_2018.pdf](https://www.wiwi.uni-siegen.de/banken/dokumente/konferenzband_carf_luzern_2018.pdf), S. 160–176.
- Lesmy, Danny/Lev Muchnik/Yevgeny Mugeran: Doyoureadme? Temporal Trends in the Language Complexity of Financial Reporting, in: SSRN Electronic Journal, 2019, doi: 10.2139/ssrn.3469073, S. 4–5.
- Müller, Stefan/Patrick Saile: ESEF: Praktische Implikationen – die Zeit läuft, in: Haufe.de News und Fachwissen, 30.09.2020, [online] [https://www.haufe.de/finance/jahresabschluss-bilanzierung/digitalisierung-von-finanzberichten-esef/praktische-implikationen-fruehzeitige-vorbereitung-ist-ratsam\\_188\\_455366.html](https://www.haufe.de/finance/jahresabschluss-bilanzierung/digitalisierung-von-finanzberichten-esef/praktische-implikationen-fruehzeitige-vorbereitung-ist-ratsam_188_455366.html) [12.01.2021].
- Nitchman, Dave: XBRL Around the World, in: XBRL International Inc, 30.09.2016, [online] <https://www.xbrl.org/xbrl-around-the-world/> [25.11.2020].
- Pfister, Jan/Sandro Venetz: Die Anwendung von XBRL in IR, in: Klaus Rainer Kirchhoff/Manfred Piwinger (Hrsg.), Praxishandbuch Investor Relations, 2., überarb. u. erw. Aufl. 2009., Wiesbaden, Deutschland: Gabler Verlag, 2009, S. 433–449.
- Roos, Marko: The Dutch Taxonomy Project and structural regulatory business reporting: impact for Statistics Netherlands (Draft), in: Website der Europäischen Kommission, [online] <https://ec.europa.eu/eurostat/documents/1001617/4411693/II-1-NL-DUTCH-TAXONOMY-PROJECT.pdf> [10.12.2020].
- Securities and Exchange Commission: About EDGAR, in: U.S. Securities and Exchange Commission, 24.08.2020, [online] <https://www.sec.gov/edgar/about> [27.12.2020].

- U.S. Securities and Exchange Commission: FINAL RULE: XBRL Voluntary Financial Reporting Program on the EDGAR System, in: SEC Website, 02.03.2005, [online] <https://www.sec.gov/rules/final/33-8529.htm> [26.11.2020].
- U.S. Securities and Exchange Commission: Inline XBRL Filing of Tagged Data, in: SEC Website, 28.06.2018, [online] <https://www.sec.gov/rules/final/2018/33-10514.pdf> [27.11.2020].
- U.S. Securities and Exchange Commission: Interactive Data to Improve Financial Reporting, in: SEC Website, 30.01.2009, [online] <https://www.sec.gov/rules/final/2009/33-9002.pdf> [26.11.2020].
- Volkswagen AG: Volkswagen Interim Report Q3 2020, in: Volkswagen AG, 29.10.2020, [online] [https://www.volkswagenag.com/presence/investorrelation/publications/interim-reports/2020/Q3\\_2020\\_e.pdf](https://www.volkswagenag.com/presence/investorrelation/publications/interim-reports/2020/Q3_2020_e.pdf) [01.12.2020].
- Wang, Ding/Min Huang/Ying Wang: Analysis of the IFRS Taxonomy Extension Mode, in: 2011 International Conference on Business Computing and Global Informatization, Jg. 2011, 2011, doi: 10.1109/bcgin.2011.121, S. 461–464.
- World Wide Web Consortium: XML Linking Language (XLink) Version 1.1, in: World Wide Web Consortium, 06.05.2010, [online] <https://www.w3.org/TR/xlink11/> [15.12.2020].
- Wu, Jia/Miklos Vasarhelyi: XBRL: A New Tool For Electronic Financial Reporting, in: Business Intelligence Techniques, 2004, doi: 10.1007/978-3-540-24700-5\_5, S. 73–92.
- XBRL Deutschland e.V.: E-Bilanz / HGB-Taxonomie Version 6.0, in: XBRL DE, [online] <https://de.xbrl.org/taxonomien/e-bilanz-hgb-taxonomie-version-6-0/> [01.01.2021].

## 12 Anhang

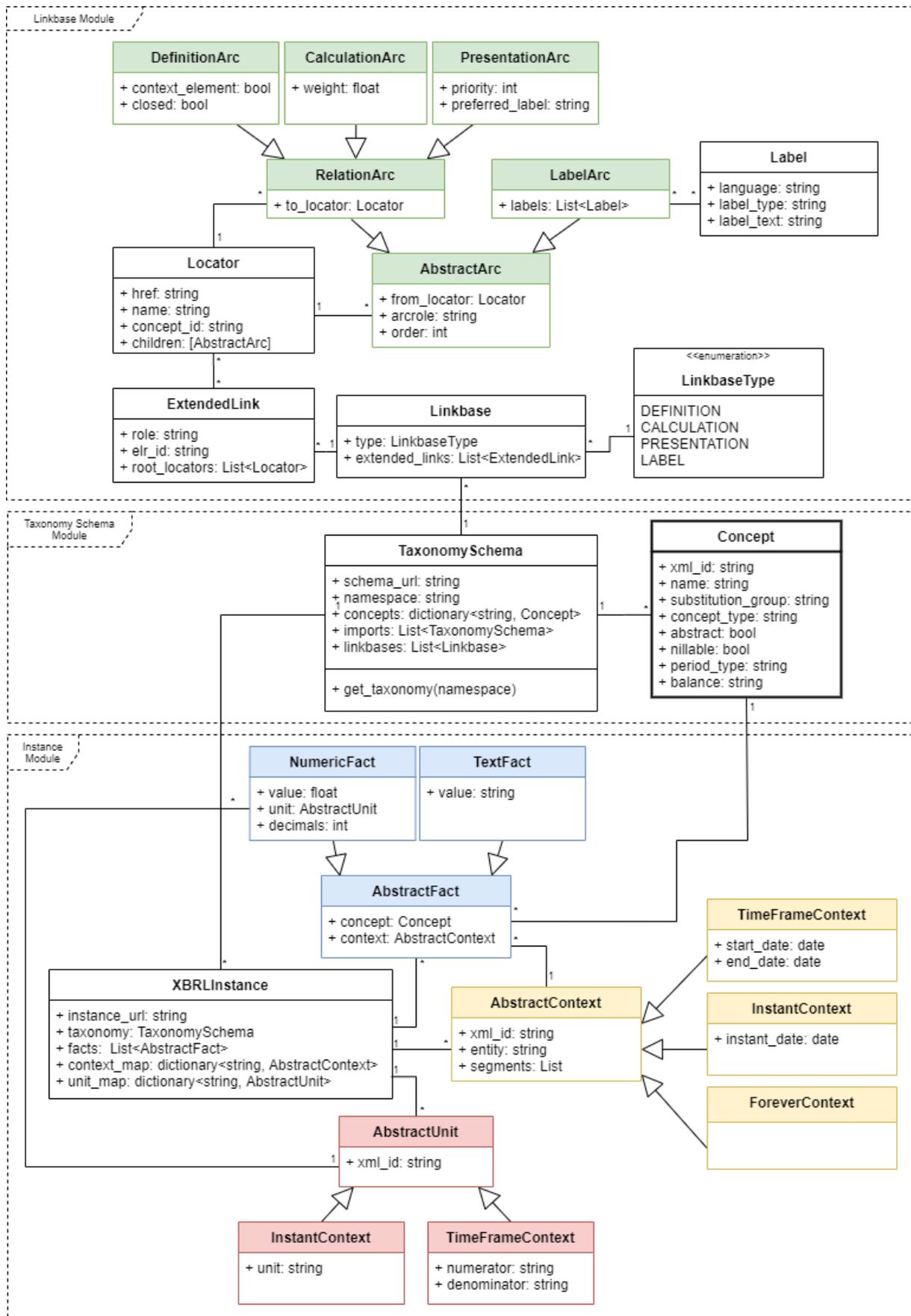


Abbildung 17: Klassendiagramm des XBRL-Parsers