



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Eine Modellierungsmethode zur Visualisierung und Analyse von Gesetzestexten“

verfasst von / submitted by

Arzo Nabizai BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Diplom-Ingenieurin (Dipl.-Ing.)

Wien, 2017 / Vienna 2017

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 066 926

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Masterstudium Wirtschaftsinformatik

Betreut von / Supervisor:

Univ.-Prof. PD Mag. Dr. Hans-Georg Fill

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Wien, 2017

Arzo Nabizai

Eidesstattliche Erklärung¹

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Wien, 2017

Arzo Nabizai

¹Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Danksagung

Mit dieser Danksagung möchte ich mich bei meinem Betreuer Hr. Prof. Hans-Georg Fill für seine unermüdliche Bereitschaft und sein Engagement bei der Unterstützung von jeglichen Fragen und Diskussionen für die vorliegende Arbeit bedanken. Seine stetige Motivation und Begeisterung für neue Ideen gaben sogar den Impuls einen gemeinsamen Beitrag für das Internationale Rechtsinformatik Symposium (IRIS) 2017 zu veröffentlichen.

Inspiration erhielt ich zudem von einigen engagierten Persönlichkeiten, die ich im Rahmen der Arbeitsgruppe "Perspektiven der Rechtssetzung" des österreichischen Parlaments kennenlernen durfte. In dieser Runde bekam ich zudem die einmalige Möglichkeit die Ergebnisse meiner Arbeit zu präsentieren und diese als Grundlage einer konstruktiven Diskussion zur Verfügung zu stellen.

Außerdem möchte ich meiner Familie und Freunden danken, die mich auf den Weg bis zur Abgabe dieser Arbeit begleitet haben, mir Mut machten dran zu bleiben und Korrektur gelesen haben. Ein besonderer Dank geht an dieser Stelle an Till Weißer.

Ich wünsche viel Freude beim Durchlesen der Arbeit.

Kurzfassung

Die Bedeutung von rechtlichen Vorgaben in Unternehmen ist enorm gestiegen, nicht zuletzt im Bereich der Informationstechnologie. Gesetzesverstöße in Form von beispielsweise der Nicht-Einhaltung von Hinweispflichten birgen dem Unternehmen Strafen von Abmahnkosten bis hin zu Schadensersatzforderungen (Knackstedt, Brelage und Kaufmann, 2006). Um die Beachtung der entsprechenden Normen in Geschäftsprozessen zu gewährleisten, bedarf es einer genauen Kenntnis der Sachlage und des Rechts. Im Besonderen muss ein Verständnis des Rechts und der entsprechenden Gesetzestexte vorhanden sein, welches gerade für Laien eine große Herausforderung darstellt.

Gesetze sind strikt aufgebaut und es existieren präzise Richtlinien, welche von Legisten bei der Arbeit mit Gesetzen eingehalten werden müssen (Bundeskanzleramt, 1990). Die daraus resultierende Struktur der Gesetze offenbart eine ganze Reihe an möglichen Abstraktions- und Gliederungsebenen, angefangen beim Abschnitt und Paragraphen bis hin zu Ziffern und Litera. Unweigerlich tritt die Frage auf, ob Gesetzestexte in ihrer textuellen Form einen adäquaten Überblick liefern und damit optimal zum Verständnis des Inhalts beitragen. Obwohl laut Richtlinie 9 "Rechtsvorschriften [...] leicht lesbar sein" (Bundeskanzleramt, 1990, S. 9) müssen, wird das Studieren einer Rechtsvorschrift weiterhin erschwert, indem eine Vielzahl von Verweise auf andere Gesetze, Paragraphen und Ziffern eine schlechtere Lesbarkeit und zusätzliche Erhöhung der Komplexität bewirken.

Gerade in der Wirtschaftsinformatik ist die Modellierung von Compliance Regelungen ein an Bedeutung gewinnender Bereich (Rinderle-Ma, Ly und Dadam, 2008). Schleicher et al. (2010) betonen, dass bei der Geschäftsprozessmodellierung der Fokus auf eine korrekte Modellierung des Geschäftsprozesses

liegt und nicht in der Beachtung von Compliance Regelungen. Genau aus diesem Grund sollte es geeignete Möglichkeiten geben, die Komplexität mit Hilfe von IT-Techniken zu verringern, dem Benutzer zugänglich zu machen und bei der Erstellung von Prozessen zu unterstützen.

In der vorliegenden Arbeit wird diese Problematik aufgegriffen und eine Lösung vorgestellt, indem eine Modellierungsmethode für die Visualisierung von Gesetzestexten entwickelt wird. Modelle dienen zur Abstraktion und Generalisierung der realen Welt (Stachowiak, 1973) und können sowohl Menschen als auch Maschinen helfen ein Themengebiet zu verstehen und zu verarbeiten. Aufgrund dieser Eigenschaften spielen sie in der Informatik in unterschiedlichsten Anwendungen eine gewichtige Rolle. Im Falle der in dieser Arbeit entwickelten Modellierungsmethode ist das Ziel, Input für Juristen und Legisten zu liefern und sie bei ihrer Arbeit mit Gesetzestexten zu unterstützen. Im Zuge dessen ist das wesentliche Konzept dieser Arbeit die Möglichkeit zur Erkennung von Qualitätsmerkmalen aus generierten Daten und Metriken sowie die Weitergabe dieser Qualitätsmerkmale in Form von praktikablen Verbesserungsvorschlägen. Aufgrund des visuellen Aspekts kann die erarbeitete Methode zudem auch die Änderung von bestehenden Gesetzen erheblich erleichtern. Das österreichische Rechtsinformationssystem (RIS), welches im Jahre 1980 entstand und seit 1997 online ist, stellt eine Dokumentation des österreichischen Rechts und eine kostenlose Informationsquelle über das Recht für Bürgerinnen und Bürger zur Verfügung. Mit der vorgestellten Lösung kann auch hier ein besserer Überblick realisiert werden, womit diese Arbeit von aktueller Relevanz und zudem im Interesse der Allgemeinheit begründet ist.

Inhaltsverzeichnis

Danksagung	v
Kurzfassung	vii
Abbildungsverzeichnis	xiii
Tabellenverzeichnis	xvii
Abkürzungsverzeichnis	xix
1. Einleitung	1
1.1. Motivation	3
1.2. Problembeschreibung	4
1.3. Forschungsfragen	5
1.4. Vorgehen	5
1.4.1. Lösungsansatz	5
1.4.2. Wissenschaftliche Methoden	6
1.4.3. Lösungskonzept	6
2. Grundlagen und verwandte Arbeiten	9
2.1. Rechtstexte in Österreich	9
2.1.1. Rechtsetzung	10
2.1.2. Aufbau von Gesetzestexten	14
2.1.3. Das Rechtsinformationssystem	17
2.2. Aspekte des Rechts in der Wirtschaftsinformatik	20

2.3.	Modellierung und Metamodellierung	22
2.3.1.	Modellierung	23
2.3.2.	Algorithmen zur automatischen Generierung von Modellen	23
2.4.	Information Retrieval im juristischen Kontext	29
2.4.1.	Natural Language Processing	30
2.4.2.	Rechtsvisualisierung und Modellierungsansätze	32
2.4.3.	Softwareprojekte im deutschsprachigen Raum	39
2.4.4.	Anwendungen in anderen Ländern	42
2.4.5.	Diskussion	43
2.5.	Konzepte zur Satzmustererkennung	46
2.5.1.	Direkter Ansatz	46
2.5.2.	Regelbasierter Ansatz	47
2.5.3.	Heuristische Methoden	48
2.5.4.	Stanford Log-linear Part-Of-Speech Tagger	49
2.5.5.	Weitere Ansätze	50
2.5.6.	Diskussion	50
3.	Implementierung einer Satzmustererkennung für Gesetzestexte	53
3.1.	Vorgehen	53
3.2.	Manuelle Analyse	54
3.3.	Automatisierte Analyse	55
3.4.	Erweiterungen und Sonderfälle	57
3.5.	Fazit	58
4.	Modellierungsmethode	59
4.1.	Metamodell der Methode	59
4.1.1.	Notation	61
4.1.2.	Syntax	63
4.1.3.	Semantik	64
4.2.	Automatische Generierung von Gesetzesmodellen	64
4.2.1.	Automatische Transformation	64
4.2.2.	Anbindung des RIS	67

4.3. Modelle	70
4.3.1. Das Gesetzesmodell	71
4.3.2. Das Konfigurationsmodell	75
5. Evaluierung	79
5.1. Evaluierungsansatz	79
5.1.1. Basis für quantitative Evaluation	80
5.1.2. Testfälle	80
5.1.3. Precision und Recall	82
5.2. Ergebnisse	83
5.2.1. Testergebnisse	83
5.2.2. Quantitative Evaluation	84
5.2.3. Precision und Recall Ergebnisse	84
6. Conclusio	87
6.1. Zusammenfassung	87
6.2. Ausblick	88
A. Abstract	91
B. Satzmustererkennung für Gesetzestexte	93
B.1. Manuelle Satzmusteranalyse	93
B.2. Automatisierte Satzmusteranalyse	108
C. Anleitung zur Verwendung der Implementierung	113
C.1. Technische Eckdaten	113
C.2. Installation	113
C.3. Bedienung	114
C.3.1. Automatische Generierung von Gesetzesmodellen	114
C.3.2. PDF-Export	115
C.4. Implementierung	115
C.4.1. Suche nach Gesetzestexten	115
C.4.2. Visualisierung von Gesetzestexten	116

C.4.3. Export von Gesetzestexten als PDF	116
D. Relevante Code-Ausschnitte	119
D.1. ADOScript	119
D.2. OGD RIS Webservice	120
Literaturverzeichnis	123

Abbildungsverzeichnis

1.1. Zuordnung der Forschungsfragen	7
2.1. Allgemeiner Prozess zur Rechtsetzung	11
2.2. Aufbau eines Gesetzes ab der Gliederungsebene des Abschnitts .	15
2.3. State-of-the-Art der Einbettung juristischer Anforderungen aus Sicht der Wirtschaftsinformatik	21
2.4. Prozess zur Informationsvisualisierung	24
2.5. Modellgenerierung	25
2.6. Beispiel zum Probabilistic Graphical Model	26
2.7. Deskriptive Statistik als Häufigkeitstabelle	27
2.8. Metamodellierungs- und Sprachebenen	28
2.9. Komponenten einer Modellierungsmethode	29
2.10. Beispiel eines Aktivitätendiagramms für die Absetzung von Pend- lerkosten	34
2.11. UML State Machine Diagramm für Gerichtsprozesse	35
2.12. Beispiel-EPK mit Elementerweiterungen für gesetzliche Bestim- mungen	37
2.13. Beispiel eines Mindmaps zur Mediation in einem Mandantenge- spräch	38
2.14. Beispiel eines Ja-oder-Nein-Entscheidungselement	39
2.15. Lebenszyklus des Gesetzes	40
2.16. "Detaildarstellung mit annotierten und markierten Argumenta- tionsmustern"	41
2.17. Beispiel einer hartkodierten Routine als Pseudocode	47

3.1. Schritt 1: Durchführung einer manuellen Analyse (§2 Abs. 1 Satz 1 bis 3 MRG)	54
3.2. Schritt 2: Durchführung einer manuellen Analyse im RIS-XML (§2 Abs. 1 Satz 1 bis 3 MRG)	54
3.3. Klassendiagramm zur Satzanalyse	56
3.4. Sequenzdiagramm	56
3.5. Ausschnitt annotierte Teilsätze im RIS-XML (§10 MRG)	57
4.1. Metamodell	60
4.2. Klassendiagramm mit Kardinalitäten	63
4.3. RIS-XML Schema	65
4.4. RIS-XML: §1 MRG Abs. 2 Z 3	66
4.5. ADOxx Architektur mit RIS Anbindung	68
4.6. Klassendiagramm für SOAP Webservice	69
4.7. Schema zu BundesnormenSearchRequest	69
4.8. Definierte Standardeinstellungen für einen SOAPRequest	70
4.9. Das Gesetzesmodell für § 13 MRG	72
4.10. Referenzen in § 10 Abs. 1 Satz 1 MRG	73
4.11. Referenz in § 1 Abs. 1 Satz 1 FAGG	73
4.12. Hinweis in § 58 Abs. 3 MRG	73
4.13. Query Ergebnisse	74
4.14. PDF-Export	75
4.15. Das Konfigurationsmodell	76
4.16. Visualisierung von Suchergebnissen	77
5.1. Evaluierungsbeispiel: manuelle Markierung	79
5.2. Evaluierungsbeispiel: automatisierte Markierung	79
5.3. Unit Testcase Equality	81
5.4. Unit Testcase Differences	81
5.5. Unit Testcase Compare To Skeleton XML	82
C.1. Aufruf des Programms zur Suche von Gesetzestexten	116

- C.2. Aufruf des Programms zur Visualisierung eines Gesetzesmodells 116
- C.3. Aufruf des Programms zum PDF-Export eines Gesetzesmodells . 117

Tabellenverzeichnis

2.1. Parlamentarischer Gesetzgebungsprozess	12
2.2. Vergleich der Modellierungsansätze	44
2.3. Vergleich der Editoren	44
2.4. Vergleich der länderspezifischen Ansätze	44
2.5. Vergleich der Ansätze nach Autoren	45
3.1. Ergebnisse der Satzmusteranalyse	55
4.1. Notation der Klassen in ADOxx - Teil 1	61
4.2. Notation der Klassen in ADOxx - Teil 2	62
4.3. Relationen in ADOxx	63
4.4. Mapping des RIS-XML zur Modellierungssprache	67
5.1. Basis für quantitative Evaluation des MRG	80
5.2. Quantitative Evaluation des MRG	84
5.3. Werte zur Berechnung von Precision und Recall	85
C.1. Technische Eckdaten zur Implementierung	113
D.1. Erstellung einer Suchmaske im Tool(ADOxx)	119
D.2. SOAP Dokumentenabfrage über OGD RIS Webservice 2.2	120
D.3. SOAP Suchabfrage über OGD RIS Webservice 2.2	121

Abkürzungsverzeichnis

ABl. Amtsblatt

Abs. Absatz

CRF Conditional Random Field

EPK Ereignis-gesteuerte Prozesskette

HTML Hypertext Markup Language

IR Information Retrieval

KSchG Konsumentenschutzgesetz

MRG Mietrechtsgesetz

NLP Natural Language Processing

OCL Object Constraint Language

OGD Open Government Data

OMG Object Management Group

RIS Rechtsinformationssystem

RIS-XML Rechtsinformationssystem Extensible Markup Language

RL Richtlinie

RTF Rich Text Format

UML Unified Modeling Language

XML Extensible Markup Language

Z Ziffer

1. Einleitung

Die Bedeutung von rechtlichen Vorgaben in Unternehmen ist enorm gestiegen, nicht zuletzt im Bereich der Informationstechnologie. Gesetzesverstöße in Form von beispielsweise der Nicht-Einhaltung von Hinweispflichten können für Unternehmen Strafen von Abmahnkosten bis hin zu Schadensersatzforderungen nach sich ziehen (Knackstedt, Brelage und Kaufmann, 2006). Um die Beachtung der entsprechenden Normen in Geschäftsprozessen zu gewährleisten, bedarf es einer genauen Kenntnis der Sachlage und des Rechts. Im Besonderen muss ein Verständnis des Rechts und der entsprechenden Gesetzestexte vorhanden sein, welches gerade für Laien eine große Herausforderung darstellt.

Gesetze sind in ihrer textuellen Form strikt aufgebaut und es existieren präzise Richtlinien, welche von Legisten bei ihrer Arbeit an und mit Gesetzen eingehalten werden müssen (Bundeskanzleramt, 1990). In Österreich sind diese in den Legistischen Richtlinien von 1990 festgesetzt. Sie geben an, wie Gesetze aufgebaut und gegliedert sein müssen. Dabei gibt es eine Vielzahl verschiedener Abstraktions- und Gliederungsebenen (vgl. Bundeskanzleramt, 1990, Richtlinie 12) - angefangen beim Abschnitt und Paragraphen bis hin zu Ziffern und Litera. Mit Blick auf diese Komplexität tritt unweigerlich die Frage auf, ob Gesetzestexte dem Leser einen adäquaten Überblick erlauben und somit zum Verständnis der Texte beitragen. Obwohl laut Richtlinie 9 "Rechtsvorschriften [...] leicht lesbar sein" (Bundeskanzleramt, 1990, S. 9) sollten, kommt hinzu, dass eine Reihe von Verweisen auf andere Gesetze, Paragraphen und Ziffern eine schlechtere Lesbarkeit und somit eine weitere Erhöhung der Komplexität bewirken.

Die Vielzahl an Verweisungen im Gesetzestext lassen sich leicht mit folgendem Gesetzesausschnitt aus § 1 Abs. 1 und 2 Z 1 und 3 und 8 FAGG zeigen. Die fett

1. Einleitung

markierten Textbestandteile sollen die Verweisungen hervorheben.

§ 1. (1) *Dieses Bundesgesetz gilt für Fernabsatz- und außerhalb von Geschäftsräumen geschlossene Verträge (Fern- und Auswärtsgeschäfte) zwischen Unternehmern und Verbrauchern (§ 1 KSchG).*

(2) *Dieses Bundesgesetz gilt – soweit in § 8 Abs. 4 nicht anderes bestimmt ist – nicht für Verträge,*

1. die außerhalb von Geschäftsräumen geschlossen werden (§ 3 Z 1) und bei denen das vom Verbraucher zu zahlende Entgelt den Betrag von 50 Euro nicht überschreitet,

[...]

*3. über Gesundheitsdienstleistungen gemäß **Artikel 3 Buchstabe a der Richtlinie 2011/24/EU** über die Ausübung der Patientenrechte in der grenzüberschreitenden Gesundheitsversorgung, **ABl. Nr. L 88 vom 4.4.2011 S. 45**, unabhängig davon, ob sie von einer Einrichtung des Gesundheitswesens erbracht werden, dies mit Ausnahme des Vertriebs von Arzneimitteln und Medizinprodukten im Fernabsatz,*

[...]

*8. die in den Geltungsbereich der **Richtlinie 90/314/EWG über Pauschalreisen, ABl. Nr. L 158 vom 23.6.1990 S. 59**, fallen,*

[...]

Ziffer 3 des Beispiels zeigt zudem den verschachtelten Satzbau, welcher für den Leser als anspruchsvoll wahrgenommen werden kann. Diese Punkte, wie der anspruchsvolle Satzbau, sowie der häufige Einsatz von Referenzen auf andere Gesetzestexte sorgen dafür, dass sich die Analyse von Gesetzestexten meist recht mühsam gestaltet und mit erheblichem Zeitaufwand verbunden ist.

Modelle dienen zur Abstraktion und Generalisierung der realen Welt (Stachowiak, 1973). Sie können sowohl Menschen als auch Maschinen helfen ein Themengebiet zu verstehen und zu verarbeiten. Im Kontext von Gesetzestexten können sie verwendet werden, um für die genannten Probleme Lösungen zu entwickeln und zu einer Vereinfachung der Analyse von Gesetzen beitragen. Mit

Hilfe von Modellen bzw. einer domänen-spezifischen Modellierungsmethode kann zudem die Änderung bestehender Gesetze erleichtert werden, wobei vor allem der visuelle Aspekt Vorteile mit sich bringt. Auch die Gestaltung neuer Gesetze kann erheblich beschleunigt und vereinfacht werden.

Gerade in der Wirtschaftsinformatik ist die Modellierung von Compliance Regelungen ein an Bedeutung gewinnender Bereich (Rinderle-Ma, Ly und Dadam, 2008). Schleicher et al. (2010) betonen, dass bei der Geschäftsprozessmodellierung der Fokus auf einer korrekten Modellierung des Geschäftsprozesses liegt und nicht in der Beachtung von Compliance Regelungen. Genau aus diesem Grund sollte es geeignete Möglichkeiten geben, die Komplexität mit Hilfe von IT-Techniken zu verringern, dem Benutzer zugänglich zu machen und bei der Erstellung von Prozessen zu unterstützen.

1.1. Motivation

Lachmayer (2015, S. 341) meint "[...] die öffentliche Diskussion über rechtliche Inhalte bereits im Gesetzgebungsprozess, aber auch nach In-Kraft-Treten der Gesetze, hängt wesentlich davon ab, ob die rechtlichen Inhalte auch verständlich sind. Ohne klare Gesetzestexte ist die Wahrscheinlichkeit einer verzerrten Diskussion über rechtliche Reformvorhaben sehr groß." Man kann also sagen, dass eine gute Legistik Hauptbestandteil einer guten Rechtsetzung sein sollte. Und auch Kölbl (2015) unterstreicht die Notwendigkeit der Verständlichkeit von Gesetzestexten für die Bürger, da diese in aller Regel von den von Rechtsexperten verfassten Texten betroffen sind. Der österreichische Verfassungsgerichtshof hat sogar mit VfSlg 3130/1956 die Unverständlichkeit von Normen aufgenommen und besagt in VfSlg 12420/1990, dass "nur mit subtiler Sachkenntnis, außerordentlichen methodischen Fähigkeiten und einer gewissen Lust zum Lösen von Denksport-Aufgaben überhaupt verstanden werden kann, welche Anordnungen hier getroffen werden sollen[.]".

Die vorliegende Arbeit greift die beschriebene Problematik auf und stellt

eine Lösung vor, indem eine Modellierungsmethode für die Visualisierung von Gesetzestexten entwickelt wird. Diese Modellierungsmethode hat zum Ziel Input für Juristen und Legisten zu liefern. Im Zuge dessen ist das wesentliche Konzept dieser Arbeit die Möglichkeit zur Erkennung von Qualitätsmerkmalen aus generierten Daten und Metriken sowie die Weitergabe der Qualitätsmerkmale in Form von praktikablen Verbesserungsvorschlägen (Strahonja, 2006; van Engers et al., 2001). Ebenso kann mit der erarbeiteten Lösung ein besserer Überblick realisiert werden, welcher schlussendlich auch Laien zu Gute kommt und somit von aktueller Relevanz und im Interesse der Allgemeinheit begründet ist.

Diese Arbeit soll eine Grundlage für Weiterentwicklungen bieten. Der Fokus liegt zunächst auf der Verarbeitung der Struktur von Gesetzen, welche im nächsten Schritt mit semantischen Ansätzen erweitert werden kann.

1.2. Problembeschreibung

Für die Umsetzung der Gesetzestexte in Modelle müssen zunächst die Grundkomponenten von Gesetzen (Abschnitt, Paragraph, Absatz, Ziffer, Litera) in Form einer Modellierungsmethode abgebildet werden. Dabei kann in zwei hauptsächliche Aspekte unterschieden werden. Zum einen die Verarbeitung der Sprache in den Gesetzestexten und zum anderen die XML-Transformationen in und aus den Modellen.

Zusätzlich müssen auch Verweise auf andere Gesetzesabschnitte oder ähnliches identifiziert, analysiert und richtig verknüpft werden. Dabei kommen laut den Legistischen Richtlinien sowohl statische als auch dynamische Verweise in Frage. Der Unterschied zwischen diesen liegt darin, dass statische auf eine "bestimmte Fassung" (vgl. Bundeskanzleramt, 1990, RL 60) einer Rechtsvorschrift und im Gegensatz dazu, dynamische auf Rechtsvorschriften "in ihrer jeweiligen Fassung" (vgl. Bundeskanzleramt, 1990, RL 61) angewendet werden. Die Erkennung von Verweisen können erst nach einer Analyse und Visualisierung der Syntax von Gesetzen behandelt und im nachfolgenden Schritt hinzugefügt

werden. Grund hierfür ist die Kenntnis der genauen Syntax zur korrekten Identifizierung von Referenzen zu anderen Gesetzesteilen.

1.3. Forschungsfragen

Die vorliegende Arbeit ist anhand von drei Forschungsfragen aufgebaut, mit Hilfe derer die Herangehensweise an die Themenstellung strukturiert wird.

1. Wie lässt sich die Gestaltung, Änderung und Analyse von Gesetzen mit Hilfe von Modellen einfacher gestalten?
2. Wie lässt sich die automatische Transformation von Gesetzestexten in Modelle (und vice versa) mit Hilfe einer Modellierungsmethode realisieren?
3. Welche Evaluierungsmöglichkeiten ergeben sich somit über die Qualität und Komplexität von Gesetzestexten?

1.4. Vorgehen

In diesem Kapitel soll das Vorgehen kurz umrissen werden. Dazu wird der Lösungsansatz, die wissenschaftlichen Methoden und das Lösungskonzept beschrieben.

1.4.1. Lösungsansatz

Die vorliegende Problematik soll mit Hilfe einer auf rechtliche Bedingungen basierenden, hergeleiteten Modellierungsmethode gelöst werden. Da im Speziellen die Wirtschaftsinformatik als Katalysator zwischen Unternehmen und Technologie gesehen wird (Zitat von D. Karagiannis), soll anhand dieser Methode eine Möglichkeit geschaffen werden eine Vielzahl an Gesetzestexten analysieren und visualisieren zu können. Diese Methodik unterstützt somit das zwischen einem Großteil der Professoren der DACH-Region vereinbartem und im European

Journal of Information Systems publiziertem Memorandum zur gestaltungsorientierten Wirtschaftsinformatik, in dem es auch heißt, dass Design "Artifacts should be created through generally accepted methods, be justified as much as possible and be contrasted with solutions already known in science and business[.]" (Österle et al., 2011, S. 9). Gerade für Unternehmen stellt dies besonders bei der Abstimmung von Unternehmensstrategien und Informationverarbeitung (Mertens et al., 2012) hilfreiche Methoden zur Verfügung.

1.4.2. Wissenschaftliche Methoden

Wilde und Hess ("Forschungsmethoden der Wirtschaftsinformatik - Eine empirische Untersuchung") nennen sechs Kernmethoden für wissenschaftliche Arbeiten der Wirtschaftsinformatik. Folgende zwei Methoden werden in dieser Arbeit zum Einsatz kommen.

Formal-/konzeptionell- und argumentativ-deduktive Analyse umfasst

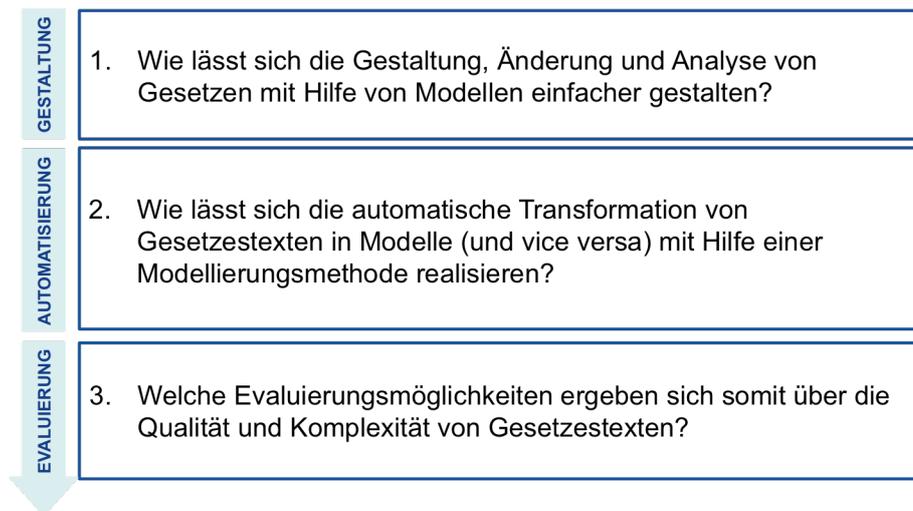
drei Methoden, von denen die konzeptionell- und argumentativ-deduktive verwendet wird, um eine konstruktive und qualitative Durchführung zu gewährleisten.

Prototyping dient als ersten Schritt dazu neue Erkenntnisse zu gewinnen, um sich dann der Entwicklung und Implementierung und anschließender Evaluierung der Modellierungsmethode zu widmen.

1.4.3. Lösungskonzept

Die vorgestellten Forschungsfragen können grob drei Oberbegriffen zugeordnet werden: der

- Gestaltung,
- Automatisierung
- und Evaluierung.



Quelle: eigene Darstellung

Abbildung 1.1.: Zuordnung der Forschungsfragen

Abbildung 1.1 zeigt die Zuordnung der Begriffe zu den Forschungsfragen. Anhand dieser Oberbegriffe kann ein Ansatz für das weitere Vorgehen entwickelt werden, welcher eine strukturierte Herangehensweise an die Problematik sicherstellt. Das momentane Vorgehen zur Unterteilung der Gesetzestexte, welches bis zur Ebene der (Sub-)Litera reicht, birgt augenscheinlich noch Potential für Verbesserungen. Zum einen finden Verweise zu anderen Gesetzen oder Gesetzesteilen auf Satzebene statt, zum Anderen werden semantische Analysen mit Hilfe von Sätzen durchgeführt. Aus diesen Gründen wird für die erste Forschungsfrage zum Ziel gesetzt, die Abbildung der Gesetzestexte auf Satzebene durchzuführen. Dieser Punkt ermöglicht einfachere und schnellere Änderungen und Analysen der vorliegenden Gesetzestexte und bietet zusätzlich die Möglichkeit zur Referenzierung auf Satzebene.

Forschungsfrage 2 beinhaltet die Entwicklung und Implementierung eines Automatisierungsansatzes, welcher aus den Erkenntnissen der vorangehenden Forschungsfrage abgeleitet wird. Dabei ist zunächst eine manuelle Analyse anhand eines Gesetzestextes durchzuführen. Als Beispiel soll hierbei das Mietrechtsgesetz (MRG) dienen. Anschließend werden Satzmuster identifiziert,

welche den Beginn eines Satzes markieren können. Aufbauend auf diesem Ergebnis folgt eine formale Abbildung der Satzmuster mit Hilfe von regulären Ausdrücken, welche zur Entwicklung einer automatisierten Satzmustererkennung dienen. Der Aspekt der Automatisierung bei dieser Arbeit beinhaltet auch die automatische Transformation von Gesetzestexten in Modelle. Da mit Hilfe der automatischen Satzmustererkennung nun die Gesetzelemente auf Satzebene definiert sind, kann die Umwandlung in Modelle auch auf dieser Ebene durchgeführt werden.

Die Ergebnisse der manuellen sowie automatisierten Analysen dienen zudem der Evaluierung. Dabei wird die manuelle und automatische Satzmustererkennung verglichen und eine Weiterentwicklung des Automatisierungsansatzes ermöglicht. Als Ergebnis können somit Analysen über die Qualität und Komplexität von Gesetzestexten durchgeführt werden.

2. Grundlagen und verwandte Arbeiten

In diesem Kapitel werden zunächst einige grundlegende, relevante Aspekte dieser Arbeit genauer erläutert. Dazu zählen die Einführungen in folgende Bereiche:

- den Grundlagen zu Rechtstexten in Österreich,
- den Aspekten des Rechts in der Wirtschaftsinformatik,
- der Modellierung und Metamodellierung,
- dem Information Retrieval im juristischen Kontext
- und den Konzepten zur Satzmustererkennung.

Im Kapitel des Information Retrieval wird neben dem Natural Language Processing auch vor allem auf die Rechtsvisualisierung und den Modellierungsansätzen im juristischen Kontext eingegangen und exemplarische Anwendungen für die Rechtsvisualisierung in anderen Ländern vorgestellt.

2.1. Rechtstexte in Österreich

Ziel der vorliegenden Arbeit ist eine visuelle Darstellung der Gesetzestexte zu erarbeiten. Für die Gestaltung dieser dient in diesem Kapitel die Vorstellung der Grundlagen für das österreichische Recht. Besonderes Augenmerk hat dabei der

Aufbau der Gesetzestexte, welcher eine gesetzlich korrekte Abbildung der Struktur von Gesetzen gewährleistet. In diesem Zusammenhang ist es von besonderer Bedeutung auf weiterführende, semantische Weiterentwicklungen hinzuweisen, welche aufbauend auf die Darstellung der Syntax von Gesetzestexten angestrebt werden können.

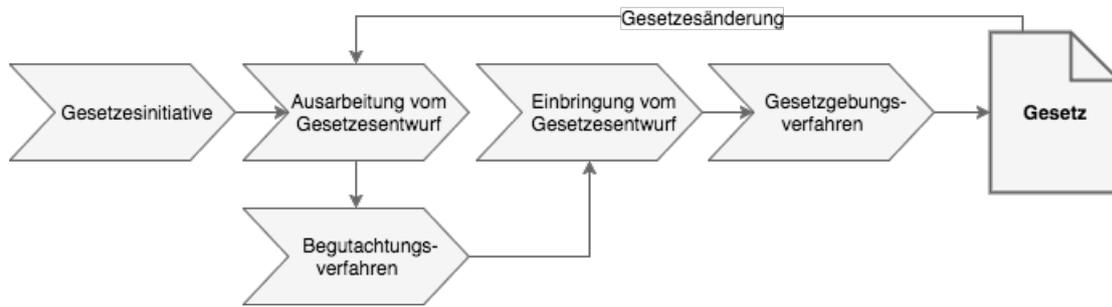
Zunächst sollen aber Grundlagen zur Rechtsetzung veranschaulicht werden. Dazu wird auch der österreichische, parlamentarische Gesetzgebungsprozess geschildert und die Rolle des Legisten hervorgehoben.

2.1.1. Rechtsetzung

Die Rechtsetzung nimmt nicht nur in Österreich, sondern auch auf der Ebene der Europäischen Union (EU) eine große Rolle als "Steuerungs- und Gestaltungsmittel" (Ismayr, 2008a, S. 9) ein. Bei diesem Prozess sind eine Reihe von Akteuren beteiligt, die sich aus den folgenden Parteien in den einzelnen Mitgliedsstaaten der EU zusammensetzen: Verfassungsorgane, Parteien, Verbände, Massenmedien, intermediäre Organisationen, regionale Institutionen sowie Kommunen (Ismayr, 2008a). "In diesem Sinne ist der Gesetzgebungsprozess als arbeitsteiliges Vorgehen unterschiedlicher Personen und Institutionen zu begreifen[.]" (Lachmayer, 2015, S. 339). Dies bedeutet, dass in diesem komplexen Prozess unterschiedliche Kompetenzen von Nöten sind, welche sich nicht nur auf rechtliche Experten beschränken, sondern auch fachliche und legistische involvieren.

Prozess zur Rechtsetzung

Ein allgemeiner Prozess zur Rechtsetzung kann sich wie in Abbildung 2.1 zusammensetzen. Den Angaben der Analysen von Ismayr (2008b) zufolge, welche sich mit der Gesetzgebung in den EU-Staaten beschäftigen, lässt sich dieser Prozess auch auf EU-Ebene verallgemeinern. Es beginnt mit einer Gesetzesinitiative, welche eine Ausarbeitung zum Gesetzesentwurf zur Folge hat. Dieser Entwurf muss durch ein Begutachtungsverfahren, bevor es als Gesetzesentwurf im Parlament eingebracht wird. Das Gesetzgebungsverfahren entscheidet, ob ein



Quelle: eigene Darstellung

Abbildung 2.1.: Allgemeiner Prozess zur Rechtsetzung

Gesetz zustande kommt. Bei Gesetzesänderungen bedarf es der Ausarbeitung der Änderungen in Form eines neuen Gesetzesentwurfes, welcher die gleichen Schritte wie nach einer Gesetzesinitiative durchläuft.

Laut § 24 des Bundesverfassungsgesetzes liegt die Zuständigkeit der Gesetzgebung in Österreich beim Nationalrat und Bundesrat. Dennoch kann es zu einem fakultativem Gesetzes- oder Verfassungsreferendum oder obligatorischem Verfassungsreferendum kommen, sodass das Bundesvolk mit einfacher Mehrheit über das Gesetz entscheidet (Pelinka, 2008). Wie dieser Prozess im Detail gestaltet ist, soll in den nächsten Abschnitten erläutert werden.

Gesetzesinitiativen

In Österreich können Gesetzesinitiativen auf viele verschiedene Arten eingereicht werden. Dazu zählen (Parlament, 2017c):

- Entwürfe, Begutachtungsverfahren oder Regierungsvorlagen der Bundesregierung,
- Anträge von Abgeordneten,
- Anträge von Ausschüssen des Nationalrates,
- Gesetzesanträge des Bundesrates
- und das Volksbegehren, den Gesetzesanträgen von BürgerInnen.

Parlamentarischer Gesetzgebungsprozess

Der Gesetzgebungsprozess in Österreich lässt sich in einen vorparlamentarischen und einen parlamentarischen Bereich unterteilen. Pelinka und Rosenberger (2003) liefern dazu einen Überblick, welche Aufgaben und Prozessschritte in welchem Bereich fallen. Die Auflistung in Tabelle 2.1 zeigt die einzelnen Schritte nach einer Gesetzesinitiative. Dabei kann man grob sagen, dass die Ausarbeitung des Gesetzesentwurfes und das Begutachtungsverfahren in den vorparlamentarischen Bereich fällt. Die Einbringung der Gesetzesvorlage findet dann im parlamentarischen Bereich statt. Hier wird im Gesetzgebungsverfahren über den Entwurf beraten, diskutiert und sowohl beim Nationalrat als auch im Bundesrat abgestimmt (Pelinka, 2008).

A. Vorparlamentarischer Bereich
1. Interessen
2. Parteien, Verbände oder Bürokratie artikulieren Interessen
3. Regierung (ein Bundesminister) gibt Auftrag an Bürokratie, einen Gesetzentwurf ("Ministralentwurf") auszuarbeiten
4. Entwurf wird ausgearbeitet
5. Begutachtungsverfahren - Interessen nehmen (nochmals) Einfluss
6. Ministerrat beschließt den Entwurf als Regierungsvorlage
B. Parlamentarischer Bereich
1. Nationalrat: Fraktionen und Ausschüsse beraten
2. Nationalrat: Plenum diskutiert und beschließt
3. Bundesrat stimmt zu oder legt ein bloß aufschiebendes Veto ein
4. Bundespräsident und Bundesregierung erfüllen die übrigen formalen Voraussetzungen

Quelle: Pelinka und Rosenberger (2003, S. 442)

Tabelle 2.1.: Parlamentarischer Gesetzgebungsprozess

Die Präsidenten der jeweiligen Ausschüsse unterschreiben die Beschlüsse, die vom Bundespräsidenten beurkundet werden (Parlament, 2017b). Nach Übermittlung aller erforderlichen Dokumente an das Bundeskanzleramt kann dieses das Gesetz kundmachen. Dies geschieht im Bundesgesetzblatt (Parlament, 2017a).

Mit dem Projekt E-Recht wurde der elektronische Rechtserzeugungsprozess eingeführt, wodurch ab Januar 2004 Bundesgesetzblätter elektronisch im Internet kundgemacht werden können (Stöger und Weichsel, 2008; Weichsel, 2014). Erre-

ichte Ziele waren vor allem Kosteneinsparungen, Schnelligkeit und Aktualität, sodass die elektronische Kundmachung die vorher papierförmigen Gesetzesblätter vollständig ablöst. Somit ist die Veröffentlichung der Gesetzesblätter über das sogenannte Rechtsinformationssystem rechtlich verbindlich.

Gesetzesänderungen

Für Änderungen bestehender Gesetze bedarf es in Österreich Novellen. In diesen legen Legisten fest, wie die Novellierungsanordnungen, d.h. die detaillierten Änderungen im Wortlaut des Gesetzes, lauten, welche dann auf die Gesetzesfassung angewendet werden. Die Anwendung der Novellierungsanordnungen auf Rechtstexte erfolgt noch heute manuell durch geschultes Personal (Reimer, 2008).

Die Rolle des Legisten

Die Aufgabe der Legistik ist es, klare und verständliche Gesetze zu formulieren, um es dem Adressaten zu ermöglichen den vorliegenden Text zu verstehen und anzuwenden (Lachmayer, 2015). "Es steht die systematische Integration des neuen Gesetzestextes im Vordergrund. Dies betrifft sowohl begriffliche Konsistenz, sprachliche Formulierungen, systematische Strukturen und einen Blick für das Gesamtkonzept, sowie die Einbettung in die Rechtsordnung insgesamt." (Lachmayer, 2015, S. 335).

Zu diesem Zwecke wurden legistische Richtlinien verfasst, welche u.a. den Aufbau der Gesetze vorgeben, aber aus verschiedensten Gründen (Zeitdruck, Quantität an Gesetzen) nicht immer eingehalten werden können (Lachmayer, 2015). Um die Einhaltung dieser zu gewährleisten und die Bedeutung der Legistik hervorzuheben, schlägt Lachmayer (2015, S. 340) drei Verbesserungsvorschläge vor:

- "[...] die Transparenz der Mitwirkenden beim Verfassen von Gesetzestexten,
- das Zur-Verfügung-Stellen einer legistischen Ausbildung und

- eine verstärkte Qualitätskontrolle der Legistik während des und nach dem Gesetzgebungsprozess."

Eine bessere und sowohl öffentliche als auch parlamentarisch gestärkte Legistik unterstützt somit den Staat als Demokratie und ermöglicht ein besseres Verständnis für das Recht und deren Gesetzestexte.

2.1.2. Aufbau von Gesetzestexten

In den Legistischen Richtlinien von 1990 ist festgesetzt, wie Gesetze aufgebaut und gegliedert sein müssen (Bundeskanzleramt, 1990). Demnach sind Gesetze und Verordnungen in Einheiten zu gliedern. Diese Einheiten sind in folgender Reihenfolge hierarchisch aufzubauen: Teil, Hauptstück, Abschnitt, Paragraph, Absatz, Ziffer, Buchstabe (Litera). Die letzten drei Gliederungsebenen Absatz, Ziffer und Buchstabe enthalten meistens die Sätze oder Teilsätze, die die eigentliche Rechtsvorschrift beinhalten, welcher anhand folgendem Gesetzesausschnitt gezeigt werden soll:

(2) Der Hauptmieter hat das Betreten des Mietgegenstandes durch den Vermieter oder die von diesem beauftragten Personen aus wichtigen Gründen zu gestatten, wobei die berechtigten Interessen des Mieters nach Maßgabe der Wichtigkeit des Grundes angemessen zu berücksichtigen sind; er hat die vorübergehende Benützung und die Veränderung seines Mietgegenstandes bei Vorliegen der folgenden Voraussetzungen zuzulassen:

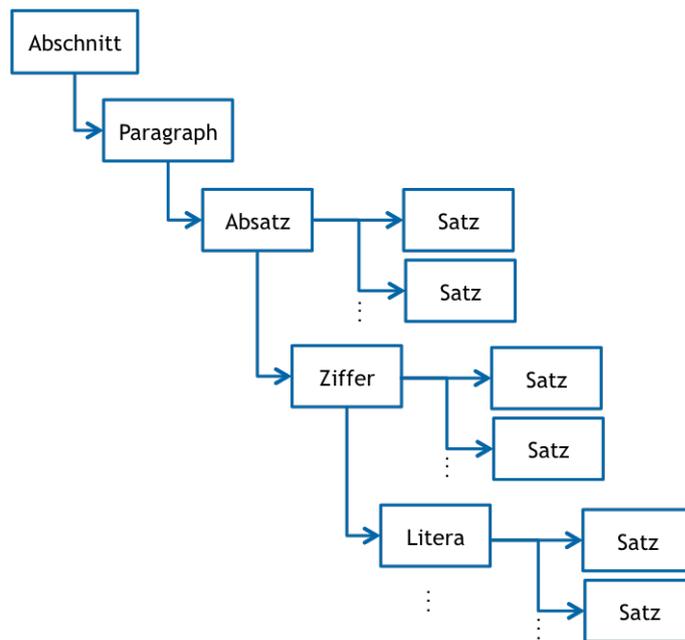
1. wenn und soweit ein solcher Eingriff in das Mietrecht zur Durchführung von Erhaltungs- oder Verbesserungsarbeiten an allgemeinen Teilen des Miethauses oder zur Behebung ernster Schäden des Hauses oder zur Erhaltung einer mitvermieteten Heiztherme, eines mitvermieteten Warmwasserboilers oder eines sonstigen mitvermieteten Wärmebereitungsgeräts in seinem oder in einem anderen Mietgegenstand notwendig oder zweckmäßig ist;

Hingegen enthalten Paragraphen in der Regel Absätze, Abschnitte Paragraphen und Hauptstücke Abschnitte. Paragraphen können aber auch Sätze enthalten, falls kein Absatz folgt. § 50 MRG lautet beispielsweise wie folgt:

§ 50. Die Gemeinden, auf die im Zeitpunkt des Inkrafttretens dieses Bundesgesetzes die Voraussetzungen des § 39 Abs. 1 zutreffen, sind durch die Kundmachung der Bundesminister für Justiz und für Inneres, BGBl. Nr. 299/1979, bestimmt.

Der Aufbau eines Gesetzes ab der Gliederungsebene des Abschnitts ist schemenhaft in Abbildung 2.2 zu sehen. Aus Gründen der Übersicht wurde dabei auf die Abbildung von Teilsätzen verzichtet. Diese können äquivalent zu den Sätzen eingeordnet werden.

Ein Beispiel für den Aufbau von Gesetzen lässt sich mit § 20 MRG zeigen. Beginnend mit dem Abschnitt der "Hauptmietzinsabrechnung" wird der Aufbau in Absatz 1 Ziffer 1 bis 2 gezeigt. Da im Vordergrund die Struktur der Gesetzestexte steht, wird aus Gründen der Übersicht der Text an dieser Stelle nicht zitiert. An dieser Stelle sind die Sätze und Teilsätzen des



Quelle: eigene Darstellung

Abbildung 2.2.: Aufbau eines Gesetzes ab der Gliederungsebene des Abschnitts

Hauptmietzinsabrechnung - Abschnitt
 § 20. (1) [...]. - Paragraph mit Absatz und Satz

1. [...] - Ziffer und Teilsatz
 - a) [...] - Litera und Teilsatz
 - b)[...]: - Litera und Teilsatz
 - aa) [...] - Sublitera und Teilsatz
 - bb) [...] - Sublitera und Teilsatz
 - cc) [...] - Sublitera und Teilsatz
 - dd) [...] - Sublitera und Teilsatz
 - c) [...] - Litera und Teilsatz
 - d) [...] - Litera und Teilsatz
 - e) [...] - Litera und Teilsatz
 - f) [...] - Litera und Teilsatz
 - g) [...] - Litera und Teilsatz
2. [...] - Ziffer und Teilsatz
 - a) [...] - Litera und Teilsatz
 - b) [...] - Litera und Teilsatz
 - c) [...] - Litera und Teilsatz
 - d) [...] - Litera und Teilsatz
 - e) [...] - Litera und Teilsatz
 - f) [...] - Litera und Teilsatz
 - g) [...] - Litera und Teilsatz

Neben diesen Gliederungsebenen sind im Besonderen Verweisungen im Gesetz hervorzuheben. Dabei handelt es sich um Verweise auf andere Gesetze, Paragraphen, Ziffern, Buchstaben, Sätze oder Teilsätze. Dabei kommen laut den Legistischen Richtlinien u.a. sowohl statische als auch dynamische Verweise in Frage. Der Unterschied zwischen diesen liegt darin, dass statische auf eine "bestimmte Fassung" (Bundeskanzleramt, 1990, RL 60) einer verwiesenen Rechtsvorschrift und dynamische hingegen auf Rechtsvorschriften "in ihrer jeweiligen Fassung" (Bundeskanzleramt, 1990, RL 61) angewendet werden.

Die Vielzahl an Verweisungen im Gesetzestext lassen sich leicht mit folgendem Gesetzesausschnitt aus § 1 Abs. 1 und 2 Ziffer 1 und 3 und 8 FAGG zeigen. Die fett markierten Textbestandteile sollen die Verweisungen hervorheben.

§ 1. (1) *Dieses Bundesgesetz gilt für Fernabsatz- und außerhalb von Geschäftsräumen geschlossene Verträge (Fern- und Auswärtsgeschäfte) zwischen Unternehmern und Verbrauchern (§ 1 KSchG).*

(2) *Dieses Bundesgesetz gilt – soweit in § 8 Abs. 4 nicht anderes bestimmt ist – nicht für Verträge,*

1. die außerhalb von Geschäftsräumen geschlossen werden (§ 3 Z 1) und bei denen das vom Verbraucher zu zahlende Entgelt den Betrag von 50 Euro nicht überschreitet,

[...]

3. über Gesundheitsdienstleistungen gemäß Artikel 3 Buchstabe a der Richtlinie 2011/24/EU über die Ausübung der Patientenrechte in der grenzüberschreitenden Gesundheitsversorgung, ABl. Nr. L 88 vom 4.4.2011 S. 45, unabhängig davon, ob sie von einer Einrichtung des Gesundheitswesens erbracht werden, dies mit Ausnahme des Vertriebs von Arzneimitteln und Medizinprodukten im Fernabsatz,

[...]

8. die in den Geltungsbereich der Richtlinie 90/314/EWG über Pauschalreisen, ABl. Nr. L 158 vom 23.6.1990 S. 59, fallen,

[...]

Diese zahlreichen Verweisungen führen zu einer schwierigeren Lesbarkeit der Gesetzestexte und bewirken somit eine Erhöhung der Komplexität (Enser et al., 1989). Bei der Gestaltung und dem Aufbau von Gesetzestexten müssen sich Legisten an die Vorgaben der Legistischen Richtlinien (Bundeskanzleramt, 1990) halten, wodurch eine Vereinheitlichung der Struktur in Form eines Schemas sichergestellt ist. Dieses Schema dient auch als Grundlage dieser Arbeit.

2.1.3. Das Rechtsinformationssystem

Das österreichische Rechtsinformationssystem (RIS), welches im Jahre 1980 entstand und 1997 online gestellt wurde, ist eine Dokumentation des österre-

ichischen Rechts und stellt eine kostenlose Informationsquelle über das Recht für BürgerInnen zur Verfügung (Bundeskanzleramt, 2016c; Stöger und Weichsel, 2008). Es erleichtert durch einfache und kostenfreie Abfragemöglichkeiten den Zugang zum Recht.

Ansätze eines Expertensystems

Enser et al. (1989) betonen die Notwendigkeit des RIS für Legisten, um diese bei der Strukturierung von Gesetzen und der "[...] Ermittlung der Auswirkungen von Novellierungen [zu] unterstützen[.]" (S. 127). Dafür ist es essentiell zu wissen oder herauszufinden, in welchen anderen Paragraphen oder Gesetzen Verweisungen zur betroffenen Novellierung existieren. Für diesen Zweck wurde von Enser et al. (1989) ein Expertensystem entworfen.

Ein Expertensystem ist nicht eindeutig definiert, da es viele verschiedene Auslegungen gibt (Jandach, 1993). Es wird allgemein als ein Computersystem angesehen, welche einer Vielzahl an Benutzern Wissen zur Verfügung stellt, das durch künstliche Intelligenz oder sogar der Simulation eines Experten (Kowalski, 1987) simuliert wird.

Das von Enser et al. (1989) entwickelte Expertensystem hat zur Aufgabe:

- Verweisungen, welche als Zitate bezeichnet werden, im Gesetzestext zu identifizieren,
- diese Verweisungen nach der Gliederungsbezeichnung aufzulösen,
- der entsprechenden Normbezeichnung zu zuordnen
- und die Normbezeichnung zu standardisieren.

Neben diesen Schritten wurde zudem eine Reihe weiterer Annahmen und Einschränkungen getroffen, die in Enser et al. (1989) nachzuschlagen sind.

Einen Überblick über juristische Expertensysteme, auch im europäischen Kreis, liefert Jandach (1993) schon 1993, welcher auch gleichzeitig darlegt wie weitreichend die Thematik ist. Die Notwendigkeit eines Expertensystems wurde

schon früh erkannt, ist allerdings im heutigen RIS nicht zu finden. Es zeigt aber, dass Bedarf besteht und die bisher erforschten Methoden in zukünftige Entwicklungen mit aufgenommen werden sollten.

Open Government Data Portal

"Werden Daten zur freien Nutzung zur Verfügung gestellt, spricht man von Open Data (OD) bzw. im Kontext von Regierungsdaten und Daten der öffentlichen Verwaltung von offenen Verwaltungsdaten bzw. Open Government Data (OGD)[.]" (Parycek und Höchtl, 2014). Die Daten des RIS sind über das Open Government Data (OGD) Portal abrufbar. Das Portal (<https://www.data.gv.at>) stellt BürgerInnen "von der Verwaltung gesammelte Daten, jedoch keine personenbezogenen" (Bundeskanzleramt, 2016d) zur freien Verfügung bereit. Seit 2012 ist es Ziel einiger österreichischer Städte und des Bundeskanzleramtes durch gemeinsame Standards den Zugriff auf Datensätze aus der Verwaltung für eine weitere Nutzung zu ermöglichen. Langfristig wird eine Zusammenarbeit im deutschsprachigen Raum Deutschland – Österreich – Schweiz – Liechtenstein (D-A-CH-LI) vorangetrieben (Bundeskanzleramt, 2016a). Das Bundeskanzleramt leistet seinen Beitrag zum OGD Portal u.a., indem es eine technische Schnittstelle zu den Daten des RIS anbietet (Weichsel, 2014).

Funktionalitäten

Neben zahlreichen weiteren Funktionalitäten stellt die Webschnittstelle des OGD zum RIS folgende Möglichkeiten bereit:

- Suche nach sämtlichen Rechtsvorschriften
- Abfrage sämtlicher Rechtsvorschriften anhand von Dokumentennummern
- Herunterladen von Rechtsvorschriften in unterschiedlichen Formaten wie z.B. HTML, XML, RTF

Die aufgelisteten Funktionalitäten wurden auch in dieser Arbeit benutzt und haben dazu beigetragen eine generelle Lösung zur Visualisierung von österreichischen Gesetzestexten zur Verfügung zu stellen.

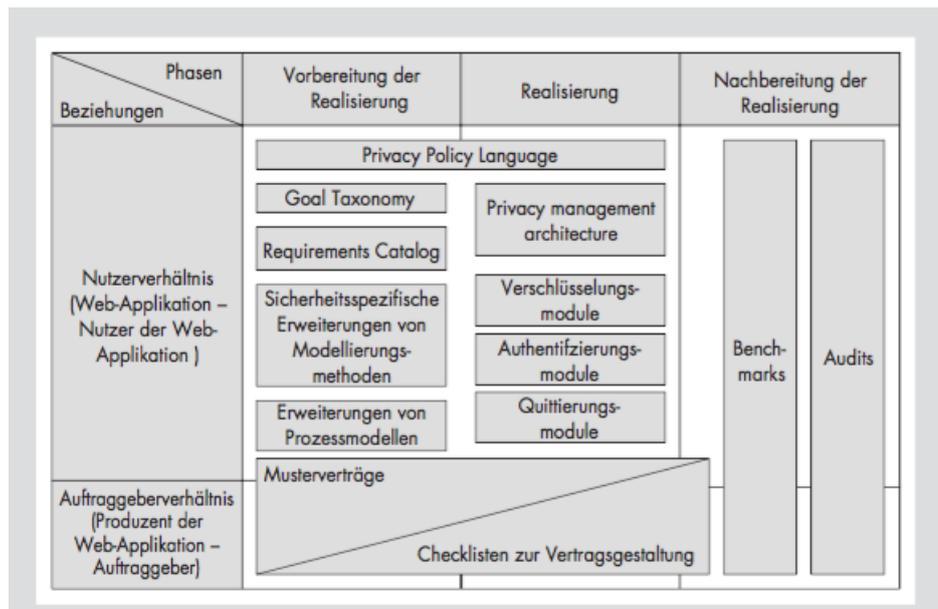
2.2. Aspekte des Rechts in der Wirtschaftsinformatik

Die Bedeutung von rechtlichen Vorgaben in Unternehmen ist enorm gestiegen, nicht zuletzt im Bereich der Informationstechnologie. Gesetzesverstöße in Form von beispielsweise der Nicht-Einhaltung von Hinweispflichten birgen dem Unternehmen Strafen von Abmahnkosten bis hin zu Schadensersatzforderungen (Knackstedt, Brelage und Kaufmann, 2006).

IT-bezogene Projekte sollten schon von Beginn an, alle relevanten Vorgaben beachten, das heißt schon bei der Konzipierung und Modellierung von Geschäftsprozessen rechtliche Rahmenbedingungen im Auge behalten. Eine nachträgliche Anpassung der Prozesse und Strukturen birgt in der Regel einen höheren Aufwand in sich. Wie in 2.4.2 vorgestellt, schlägt Olbrich (2007) vor Gesetzestexte zu modellieren, da sie für Geschäftsprozesse von großer Relevanz sind und diese häufig einschränken oder regulieren.

Knackstedt, Brelage und Kaufmann (2006) legen den Fokus auf die Einhaltung juristischer Anforderungen bei der Web-Anwendungsentwicklung und beachten dabei insbesondere die "[...] für die Wirtschaftsinformatik typische Dimension der Projektphasen [...]". In Abbildung 2.3 sieht man dazu die Zuordnung juristischer Anforderungen zu diesen Projektphasen. Dabei sind die abgebildeten Dimensionen, Phasen und Beziehungen vom Projekt dargestellt. Die Phasen teilen sie grob in die Vorbereitung der Realisierung, der Realisierung an sich und der Nachbereitung derer ein. Beziehungen vom Projekt sind zum einen die Nutzer des Projektergebnisses und der Auftraggeber des Projekt. Besonders in den ersten zwei Projektphasen bedarf es umfangreicher juristischer Überlegungen. Als Beispiel sei hier das Requirements Catalog zu nennen, welches die Anforderungen des Nutzers und das Verständnis der Entwickler vereint. Geeignete Verschlüsselungs-, Authentifizierungs- und Quittierungsmod-

ule müssen ebenfalls rechtlichen Gegebenheiten angepasst werden. Zum Auftraggeber hin müssen vertragliche Vereinbarungen getroffen werden, welche zusätzliche, juristische Unterstützung benötigt. Bei der Nachbereitung der Realisierung treffen dann Themen wie Benchmarking und Auditierung auf juristische Anforderungen.



Quelle: Knackstedt, Brelage und Kaufmann (2006, S. 31)

Abbildung 2.3.: State-of-the-Art der Einbettung juristischer Anforderungen aus Sicht der Wirtschaftsinformatik

Die dargestellten Anforderungen lassen sich auch außerhalb der Webentwicklung auf Projekte in Unternehmen anwenden und bieten einen Überblick in welchen verschiedenen Bereichen Aspekte des Rechts relevant sind.

Betrachtet man beispielsweise das Mietrecht fallen einem einige Anwendungsfälle des Rechts ein, die sowohl der Mieter, der Vermieter und die Hausverwaltung beachten müssen. Um nur ein paar Beispiele zu nennen, seien hier die Festsetzung der Mietkaufzinsen und die rechtlichen Rahmenbedingungen eines unbefristeten Mietvertrags anzubringen. In diesem Fall übernimmt die Hausverwaltung die Vermittlung eines Mietobjekts und die Übersetzung der

zugrundeliegenden Rechtsvorschriften in den Mietvertrag. Die Hausverwaltung als wirtschaftliches Unternehmen sollte dafür über geeignete Mittel und Kenntnisse verfügen, um diese Bedingung zu erfüllen. Ein Tool, welches einem in diesem Fall unterstützt, um Gesetzestexte zu visualisieren, zu erklären und zu verdeutlichen, wäre hierbei von besonderem Vorteil.

Awad und Weske (2010) betonen neben der Rolle der Gesetzgebung in Geschäftsprozessmodellen besonders die Wichtigkeit von Compliance Regelungen und binden diese als Elemente ein. Der Fachbegriff in diesem Bereich bezeichnet sich als Business Process Compliance (BPC) (Rinderle-Ma, Ly und Dadam, 2008) und wird in der Geschäftsprozessmodellierung einer immer höher werdenden Bedeutung zugewiesen. Gerade in diesem Bereich ist es für Unternehmen schwierig Compliance Regelungen adäquat zu beachten, vor allem bei der Modellierung. Diese Regelungen können in Form von Reglementierungen, Richtlinien oder Qualitätsauflagen vorliegen (Awad, Decker und Weske, 2008). Schleicher et al. (2010) betonen, dass bei der Geschäftsprozessmodellierung der Fokus auf eine korrekte Modellierung des Geschäftsprozesses liegt und nicht in der Beachtung von Compliance Regelungen. Genau aus diesem Grund sollte es geeignete Möglichkeiten geben, die Komplexität mit Hilfe von IT-Techniken zu verringern, dem Benutzer zugänglich zu machen und bei der Erstellung von Prozessen zu unterstützen.

Einer der Kernbereiche der Wirtschaftsinformatik ist die Konzeption, Implementierung und zur Verfügungsstellung von Informations- und Kommunikationstechnologie, welche das Wirtschaften im Unternehmen vereinfacht und unterstützt. Eine Modellierungsmethode zur Visualisierung und Analyse von Gesetzestexten liegt demnach im Sinne der Wirtschaftsinformatik und ermöglicht es den Anwendern Gedankengänge und Prozesse zu veranschaulichen.

2.3. Modellierung und Metamodellierung

Ein wesentlicher Bereich der Wirtschaftsinformatik ist die Modellierung von Datenstrukturen zur Visualisierung von komplexen Gegebenheiten. Gleichzeitig

ist eines der Ziele dieser Arbeit die Entwicklung und Implementierung einer Modellierungsmethode für Gesetzestexte, weshalb im Nachfolgenden genauer auf die Definitionen und Aspekte der Modellierung und Metamodellierung eingegangen wird.

2.3.1. Modellierung

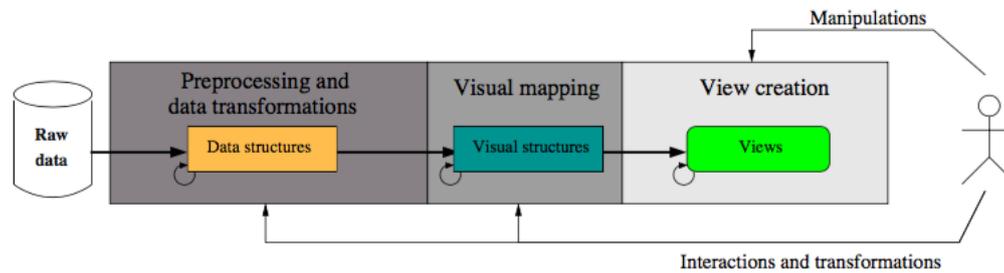
Sei es in der Softwareentwicklung oder in der Architektur - die Modellierung nimmt eine immer bedeutendere Rolle in vielen Bereichen ein. Betrachtet man die verschiedenen Einsatzbereiche und Aspekte der Modellierung detaillierter, lässt sich ein Trend zur Visualisierung von abstrakten oder komplexen Gegebenheiten feststellen.

Modelle dienen zur Abstraktion und Generalisierung eines Ausschnitts der realen Welt (Stachowiak, 1973). Sie können sowohl Menschen als auch Maschinen helfen ein Themengebiet zu verstehen und zu verarbeiten und nehmen aufgrund dieser Eigenschaften auch in der Informatik eine bedeutende Rolle ein. Generell kann man über die Modellierung zusammenfassen, dass relevante Komponenten abstrahiert, die Komplexität verringert und somit ein allgemeiner Überblick über die Problemstellung geboten wird (Kramer, 2007).

2.3.2. Algorithmen zur automatischen Generierung von Modellen

Forschungsfrage 2 dieser Arbeit befasst sich mit der automatischen Transformation von Gesetzestexten zu Modellen und vice versa. Gesetzestexte können als Daten angesehen werden, aus denen ein geeignetes Modell generiert werden soll. Um aus einem vorhandenen Daten- und Informationsbestand ein adäquates Modell automatisch zu generieren, wurde zunächst eine Analyse verschiedener Herangehensweisen und Algorithmen durchgeführt.

Ein Bereich, der sich wesentlich mit dieser Problematik beschäftigt, ist durch die Informationsvisualisierung gegeben, welche zur Aufgabe hat abstrakte Daten



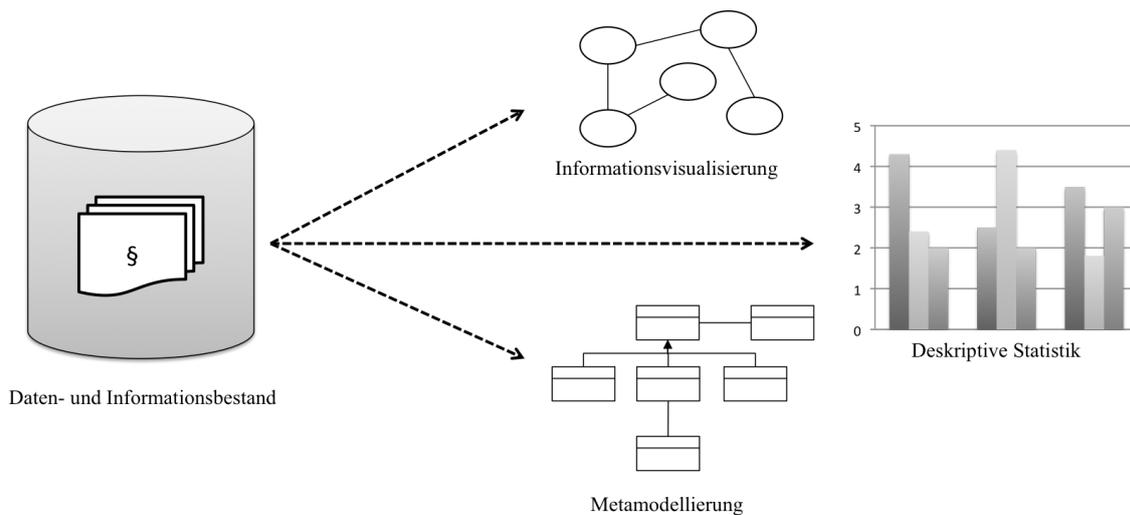
Quelle: Mazza (2009, S. 18)

Abbildung 2.4.: Prozess zur Informationsvisualisierung

in einer menschenlesbaren Form darzustellen. Card, Mackinlay und Shneiderman (1999) definieren den Begriff Informationsvisualisierung mit “the use of computer-supported, interactive, visual representations of data to amplify cognition[.]” (Card, Mackinlay und Shneiderman, 1999, S. 579). Dabei ist es wichtig dem Betrachter die Möglichkeit zu geben schnell und einfach den Sachverhalt oder die beschriebene Problematik zu verstehen, indem statt einer textuellen oder mündlichen Beschreibung eine visuelle benutzt wird (Mazza, 2009).

Der Prozess der Informationsvisualisierung lässt sich nach Mazza (2009) in drei Schritten zusammenfassen: der Aufbereitung und Datentransformationen der Rohdaten, der visuellen Zuordnung sog. Mapping und der Erstellung einer Ansicht (siehe Abbildung 2.4). In den Phasen eins und zwei interagiert, in Phase drei manipuliert der User mit seinem Input.

Die Problematik besteht nun darin aus einer Vielzahl an Daten Informationen zu extrahieren und diese zu visualisieren, um beim Benutzer Wissen zu generieren und zu entwickeln. Dazu werden in diesem Kapitel ausgewählte Bereiche und Möglichkeiten zu dieser Thematik vorgestellt und für das Ziel der vorliegenden Arbeit evaluiert. Zur Veranschaulichung der Problematik dient Abbildung 2.5, welches auch eine Zusammenfassung der im Folgenden erläuterten Möglichkeiten bietet (Fill, 2009). Aus dem Bereich der Informationsvisualisierung sollen Probabilistic Graphical Models vorgestellt werden, zudem ausgewählte Modelle der deskriptiven Statistik und die Metamodellierung.



Quelle: eigene Darstellung

Abbildung 2.5.: Modellgenerierung

Probabilistic Graphical Models

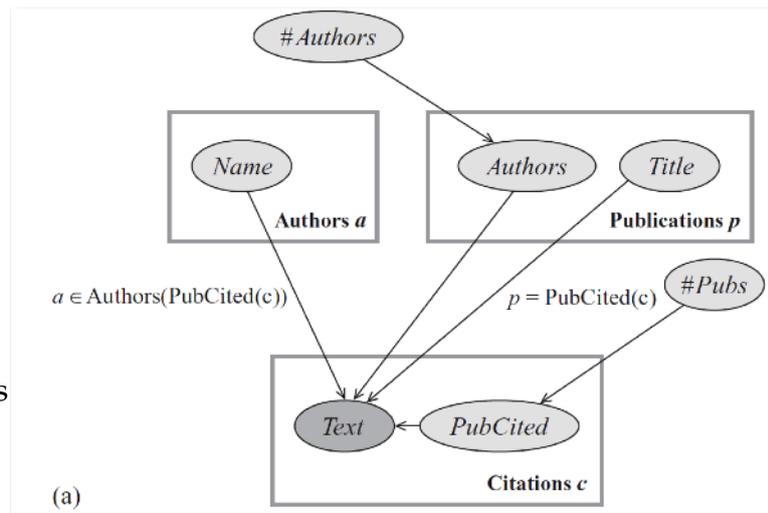
Es gibt zahlreiche Techniken der Informationsvisualisierung, darunter beispielsweise „Probabilistic Graphical Models“. Diese Technik erzeugt aus unbekanntem Daten einen Graphen mit Zufallsvariablen als Knoten, welche mit Kanten verbunden sind, falls sie voneinander abhängig sind (Koller und Friedman, 2011). Probabilistic Graphical Models benutzen Graphen zur Visualisierung, um eine kompakte Darstellung von komplexen Verteilungen zu gewährleisten. Besonders oft werden diese Modelle in Zusammenhang mit Expertensystemen in Erwähnung gebracht. Frühe Impulse gab dabei schon das Werk von Pearl (1988).

Zu den Probabilistic Graphical Models gehören sowohl Bayessche als auch Markov Netzwerke, welche in Expertensystemen zur Wissensrepräsentation verwendet werden. Das Bayessche Netz stellt einen gerichteten azyklischen Graphen dar, dessen Knoten Variablen repräsentieren, gerichtete Kanten Relationen zwischen den Variablen darstellen und mit Hilfe von Wahrscheinlichkeiten eine Gewichtung der Kanten erzeugt werden kann (Pearl, 1988). Markov Netzwerke haben hingegen ungerichtete Kanten.

Im juristischen Kontext benutzen beispielsweise Saravanan, Ravindran und

Raman (2006) Graphical Models, um Zusammenfassungen für Rechtstexte automatisiert zu erstellen. Dabei versuchen sie sich an einer neuen Methode der Conditional Random Field (CRF), wobei es sich um eine grafische Technik handelt, die einen Text anhand von gegebenen Wörtern segmentiert.

Allerdings verwenden sie CRF nicht zur Segmentierung mit Hilfe von Wörtern, sondern mit Hilfe von Sätzen, um als Ergebnis ihrer Methode eine Zusammenfassung des vorliegenden Gesetzestextes zu liefern. In einem sogenannten "distribution model" (Saranvanan, Ravindran und Raman, 2006, S.52)



Quelle: Koller und Friedman (2009)

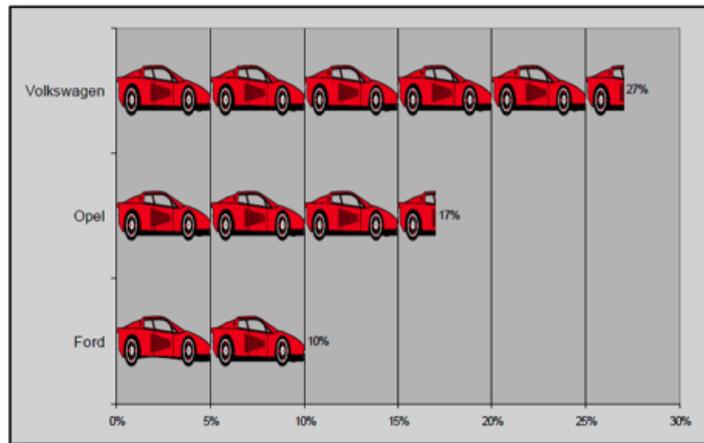
Abbildung 2.6.: Beispiel zum Probabilistic Graphical Model wurden der identifizierten Anzahl Wahrscheinlichkeiten zugewiesen, welche, nach einer bestimmten Ordnung sortiert, die Identifizierung der wichtigen Sätze in einem Rechtsdokument ermöglichen. Abbildung 2.6 zeigt ein beispielhaftes Probabilistic Graphical Model von Koller und Friedman (2009), welche eine in der Forschung häufig vorkommende Zitierung von Autoren mit ihren jeweiligen Publikationen darstellt.

Modelle der deskriptiven Statistik

Große Datenmenge lassen sich ebenso mit bekannten Methoden der deskriptiven Statistiken visualisieren. Eine Quantifizierung der Häufigkeiten in einem bestimmten Datensatz lässt sich mit Hilfe von Häufigkeitstabellen oder Histogrammen darstellen. Eine Häufigkeitstabelle listet Ergebnisse der absoluten und relativen Häufigkeit einer Stichprobe auf, welche in verschiedensten Diagrammen

ausgegeben werden können. Als Beispiele seien hier Kreis- oder Stabdiagramme angebracht. Häufigkeitsverteilungen können dann in Histogrammen dargestellt werden.

Mit Hilfe von Tools wie z.B. dem Statistik-Programm R (<https://www.r-project.org>) können Modelle der deskriptiven Statistik auf Datensätze angewendet werden. Dieses liefert dann die programmierten Visualisierung für den Anwender und ermöglicht dadurch Auswertungen durchzuführen. Hudec (2007) stellt beispielsweise in einem Graphen die Anteile verschiedener Autohersteller in einem Datensatz dar (siehe Abbildung 2.7).



Quelle: Hudec (2007, S. 21)

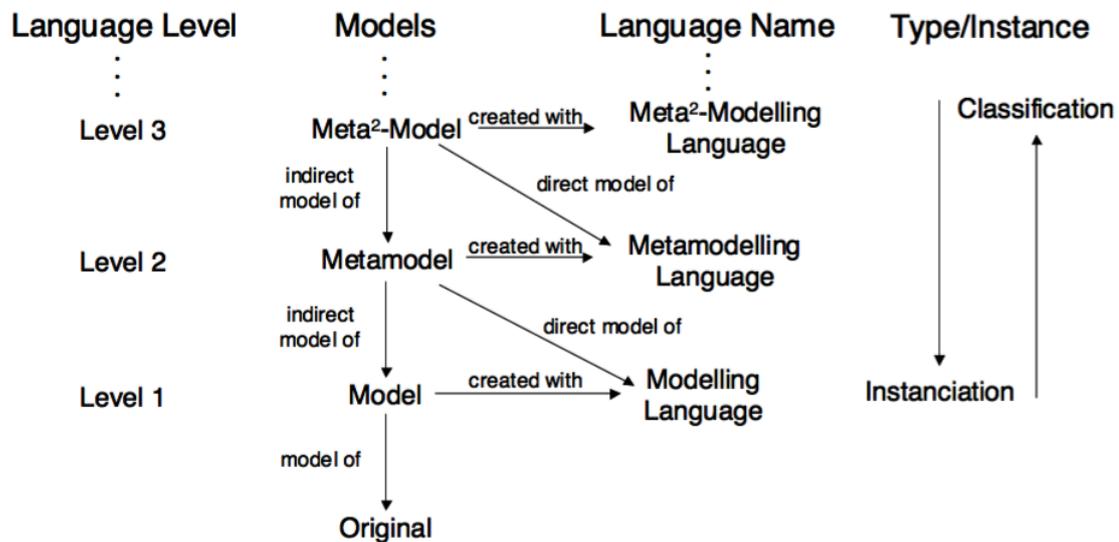
Abbildung 2.7.: Deskriptive Statistik als Häufigkeitstabelle

Meist liegen die darunterliegenden Daten in Tabellenform vor, das heißt, dass diese bereits nach bestimmten Kriterien sortiert und zugeordnet sind.

Metamodellierung

Ein weiterer Ansatzpunkt ist, die vorhandenen Daten in einem Metamodell zusammenzufassen und anhand dessen visuelle Modelle automatisch zu generieren (Falleri et al., 2008). Dieses Metamodell gibt somit die Modellstruktur und einen Teil der Semantik der Elemente vor. In der Informationsvisualisierung als auch in der deskriptiven Statistik ist hingegen typischerweise kein Metamodell vorhanden.

Ein Metamodel beschreibt die Modellierungsmethode und definiert somit eine formale Abbildung des Modells dieser Methode. Die Entwicklung eines Metamodells geschieht auch unter Zuhilfenahme einer Modellierungsmethode



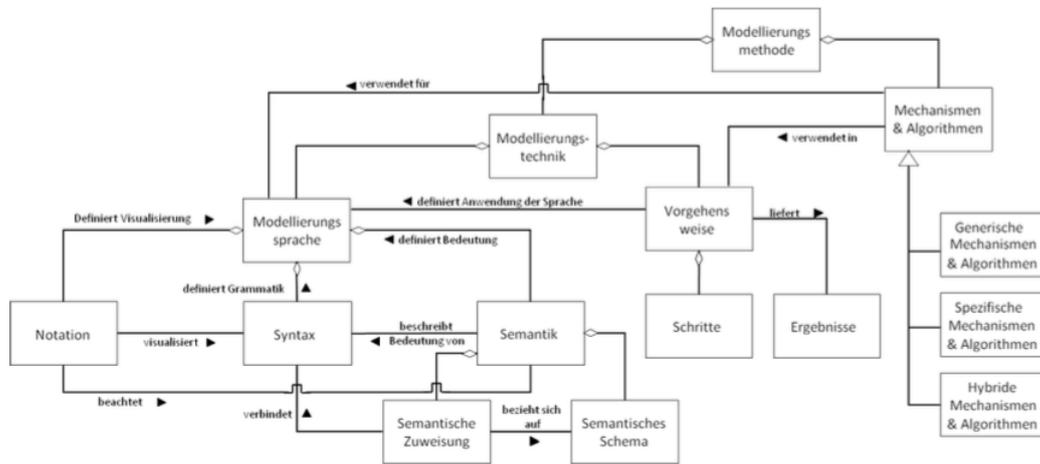
Quelle: Karagiannis und Kühn (2002, S. 4)

Abbildung 2.8.: Metamodellierungs- und Sprachebenen

und wird Metamodellierungssprache genannt. Das Modell, welches die Metamodellierungssprache beschreibt, wird Meta-Metamodell oder Meta2-Modell genannt. Diese Abstrahierungen in verschiedene Ebenen können theoretisch beliebig weitergeführt werden. In der Praxis ist es jedoch üblich eine Metamodell-Architektur mit vier Ebenen zu nutzen. Abbildung 2.8 veranschaulicht diesen Aufbau. Um eigene Modelle entwickeln zu können, kann der Metamodellierungsansatz von Karagiannis und Kühn (2002) herangezogen werden. Dabei wird eine Modellierungsmethode in zwei Komponenten geteilt: der Modellierungstechnik Komponente sowie eine Komponente, welche Mechanismen und Algorithmen beinhaltet. Die Modellierungstechnik besteht wiederum aus einer Modellierungssprache und einer Modellierungsvorgehensweise. Zur Modellierungssprache zählen die Notation, die Syntax und die Semantik. Dabei stellt die Notation die Visualisierung, die Syntax die Grammatik und die Semantik die Bedeutung einer Modellierungsmethode dar.

Die Mechanismen und Algorithmen dienen der Bearbeitung und Evaluierung der Modelle einer Modellierungsmethode. Abbildung 2.9 zeigt einen Überblick

über alle Komponenten einer Modellierungsmethode.



Quelle: Karagiannis und Kühn (2002, S. 4)

Abbildung 2.9.: Komponenten einer Modellierungsmethode

Im vorliegenden Beitrag liegen die Vorteile eines Metamodells für eine Entwicklung eines domänenspezifischen Algorithmus auf der Hand. Zum einen ist durch den Aufbau von Gesetzestexten die Struktur klar vorgegeben, zum anderen muss gewährleistet werden, dass dieselben Gesetze gleich aufgebaut sind. Aufgrund dessen wird die automatische Generierung der Modelle mit Hilfe eines eigens entwickelten Algorithmus umgesetzt (Nabizai und Fill, 2017). Die Verbindung von Syntax und Semantik der Modelle kann in Anlehnung an Fill (2015) stattfinden.

2.4. Information Retrieval im juristischen Kontext

Ein bedeutender Bereich, mit dem sich Forscher auseinandersetzen, ist der des Information Retrieval, welche die Verarbeitung von Daten und die Extrahierung von Informationen fokussiert. Die Definition des Begriffs lautet nach Manning, Raghavan und Schütze (2009): "Information retrieval (IR) is finding material

(usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)[.]. Im juristischen Kontext wird versucht Methoden des *Legal Information Retrievals* einzusetzen, um das Recht effizienter zu verstehen bzw. Rechtstexte geeignet einzusetzen.

2.4.1. Natural Language Processing

Natural Language Processing (NLP) ist die maschinelle Verarbeitung von natürlicher Sprache. Dabei geht es um die Identifizierung und Einordnung von Sprachkonstrukten, sodass Maschinen in der Lage sind geschriebene, natürliche Sprache zu verstehen. NLP Methoden können in

- lexikalische,
- syntaktische und
- semantische Analysen aufgeteilt werden (Kordjamshidi, 2013).

Lexikalische Analysen zielen darauf ab den in natürlicher Sprache verfassten Text in Abschnitte, Sätze und Wörter zu unterteilen. Diese Wörter werden nach ihren Stämmen analysiert und von Satzzeichen getrennt (Jurafsky und Martin, 2000). Anschließend übernehmen syntaktische Analysen die Zuordnung der Wörter zu Satzbauteilen, sog. Wörterklassen wie Nomen, Verben, Adjektiven oder Adverbien, usw.. Semantische NLP Methoden analysieren die Bedeutung dieser Wörter, sowohl die individuelle als auch im Kontext des Satzes.

In Hinblick auf Rechtstexte gab es mit Hilfe von NLP schon einige Ansätze. Van Gog und Van Engers (2001) setzen auf den Ansatz mit Hilfe von NLP die natürliche Sprache der Gesetzestexte in eine formale Sprache wie UML/OCL umzuwandeln. Dabei konzentrierten sie sich einerseits auf die Identifizierung von Nomen und der Modellierung von "elementary sentences" (S. 562). Zudem standen ihnen dank eines niederländischen Forschungsprojekts sogenannte

Pattern zur Verfügung, welche häufig im niederländischen Recht eingesetzt wurden.

In ihren Arbeiten beschäftigen sich auch Schweighofer und Winiwarter (1993); Merkl, Schweighofer und Winiwarter (1994); Schweighofer (1999) und Schweighofer, Winiwarter und Merkl (1995) mit heuristischen NLP Verfahren zur Abbildung von Gesetzen.

Schweighofer, Winiwarter und Merkl (1995) meinen traditionelle Information Retrieval Systeme sind den Anforderungen eines Anwalts, bei welchem es sich um einen Rechts-/Fachexperten handelt, nicht ausreichend. Grund hierfür nennen sie, dass die rechtlichen Daten stets nur in ihrer syntaktischen Struktur dargestellt werden. Das Problem für die Benutzbarkeit der Systeme in professionellen Kreisen sehen sie darin, dass alleine schon das Aneignen von Wissen ein Hindernis darstellt. Für ihren Lösungsansatz wählen sie einen statistischen Ansatz basierend auf neuronalen Netzwerken, um mit Hilfe von Textmustern das Expertenwissen zu formalisieren und somit den rechtlichen Text zu analysieren. Ähnlich ist auch der statistische Ansatz zur semantischen Erschließung von Rechtstexten von Merkl, Schweighofer und Winiwarter (1994) aufgebaut. Innerhalb dessen wird ein neuronales Netzwerk benutzt, genauer gesagt ein "self-organising map" (S. 4). Schweighofer und Winiwarter (1993) haben sogar ein Rechtsexpertensystem KONTERM entwickelt, welches den Fokus auf eine automatische Darstellung der Struktur und Inhalt eines Rechtsdokuments legt. Dabei stand eine auf Wörter basierte Analyse im Vordergrund, in der auch Abkürzungen und Sonderzeichen beachtet wurden. Allerdings ist auch hier die Methode "[...] based on statistical analysis techniques [...]" (S. 2) und daher zwar in großen Informationssystemen einsetzbar, aber heuristische Methoden erscheinen selbst bei geringer Fehlerwahrscheinlichkeit für die Bereitstellung von rechtlich verbindlichen Informationen wie im RIS weniger geeignet.

Semantische Analysen im juristischen Kontext sind in vielen Forschungsbereichen zu finden. Zahlreiche Autoren haben mit NLP Techniken die Rekonstruktion und automatisierte Extrahierung von semantischen Argumentsketten

für rechtliche Fälle vorangetrieben und dabei verschiedene Vorgehensmodelle gewählt (Wyner et al. (2010); Walter (2009)).

2.4.2. Rechtsvisualisierung und Modellierungsansätze

Im Bereich der Rechtsvisualisierung beschäftigt man sich u.a. mit Modellen und der Modellierung im Recht. Unter Rechtsvisualisierung versteht man neben zahlreichen weiteren Definitionen auch die Darstellung der Syntax und/oder Semantik von Rechtsnormen (Bergmans, 2009). Die Rechtsvisualisierung ist ein Bereich, welcher von Interdisziplinarität geprägt ist. Sie umfasst dabei nicht nur die Rechtsinformatik, sondern auch die Kommunikationswissenschaft, die Psychologie und Informatik bzw. Wirtschaftsinformatik (Schoormann und Knackstedt, 2016). Eine Verbindung zu Letzterem lässt sich beispielsweise bei Mietrechtsgesetzen herstellen. Hier ist es von besonderer Bedeutung dem Mieter als auch dem Vermieter bei der Vermietung von Objekten die Rechtslage näher zu bringen. Eine Rolle dabei könnten Makler und Mietagenturen spielen, welche sich um die Unterstützung bei der Vermittlung der Gesetzeslage an den Kunden kümmern. Näheres über die Aspekte des Rechts in der Wirtschaftsinformatik ist in 2.2 zu finden.

Schon im Mittelalter benutzte man in historischen Rechtsbüchern Bilder zur Veranschaulichung von Rechtslagen. Diese spezialisierten sich allerdings auf die Semantik der Rechtsnormen, welche ohne Zuhilfenahme eines Experten und dessen Wissen nicht aussagekräftig wäre. Zahlreiche weitere semantische Ausprägungen der Rechtsvisualisierung sind zu beobachten, wie z.B. mit Hilfe von Comics (Hahn, Mielke und Wolff, 2013; Schoormann und Knackstedt, 2016) oder Flow-Charts (Tobler und Beglinger, 2010). Neben der semantischen Visualisierung, welche zumeist einen Anwendungsfall des Gesetzestextes beschreibt (Walser Kessel et al., 2016), existiert auch eine syntaktische Visualisierung, welche sich mit der Darstellung der Struktur von Normen auseinandersetzt.

Verschiedenste Gründe führen dazu, dass es bisher zahlreiche Ansätze gibt das Recht zu visualisieren oder modellieren. Besonders lange oder stark ver-

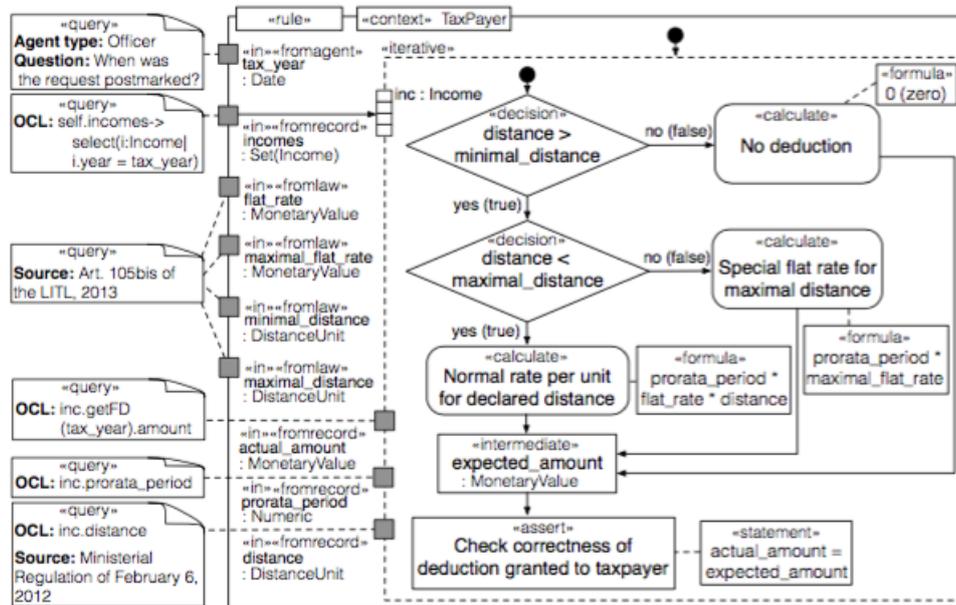
schachtelte Sätze, komplizierte Verweisungen auf andere Gesetze/-steile oder sogar Widersprüche erschweren dem Laien das Verständnis von Gesetzestexten (Kahlig, 2008).

Bisher greifen viele Forschungen auf bekannte Modellierungsmethoden zurück, aber auch eigens entwickelte Modellierungsansätze werden vorgeschlagen. Eine unterstützende Rolle bei der Entwicklung von Modellierungsansätzen spielt nicht zuletzt auch NLP. In diesem Kapitel sollen ein paar Beispiele darstellen, welche verschiedenen Möglichkeiten es zur Rechtsvisualisierung gibt.

UML

Die Unified Modeling Language (UML) genießt besonders in der Softwareentwicklung eine große Popularität. UML bietet eine Auswahl an Standard Modellierungselementen, die weit verbreitet und bekannt sind, wodurch es vielen verschiedenen Anwendern ermöglicht wird, das gleiche Modell zu verstehen und sogar zu erweitern. Auch im juristischen Kontext wird versucht diesen Vorteil auszunutzen und mit Hilfe von UML das Recht für verschiedenste Zwecke zu modellieren. Um die semantische als auch die syntaktische Visualisierung zu verbinden, greift Engesser (2014) auf UML zurück. Er begründet die Wahl von UML für Gesetzestexte damit, dass man mit Hilfe von UML und seiner Meta-Ebene auch auf die Ebenen der Sprache abstrahieren kann. Damit ist die Aufsplittung der Sprache nach der Gattung, Art, Eigentümlichkeit, Artunterschied und die zufällige Eigenschaft gemeint, welche bei der Aussagenlogik des Rechts für die Aussagenweisen stehen (Schneider und Schnapp, 2006). Die Ergebnisse der Aufsplittung können dann in die Meta-Ebene der UML Modellierungsmethode eingebunden werden, was sowohl syntaktische als auch semantische Möglichkeiten eröffnet. Auch Soltana et al. (2014) entscheiden sich für UML als Modellierungsansatz und entwickeln ein UML Profil, um das Recht mit Aktivitätendiagramme (siehe Abbildung 2.10) modellieren zu können. Darauf aufbauend transformieren sie diese Modelle dann in die Object Constraint Language (OCL), um Compliance Analysen mit Hilfe von logik-basierenden

2. Grundlagen und verwandte Arbeiten



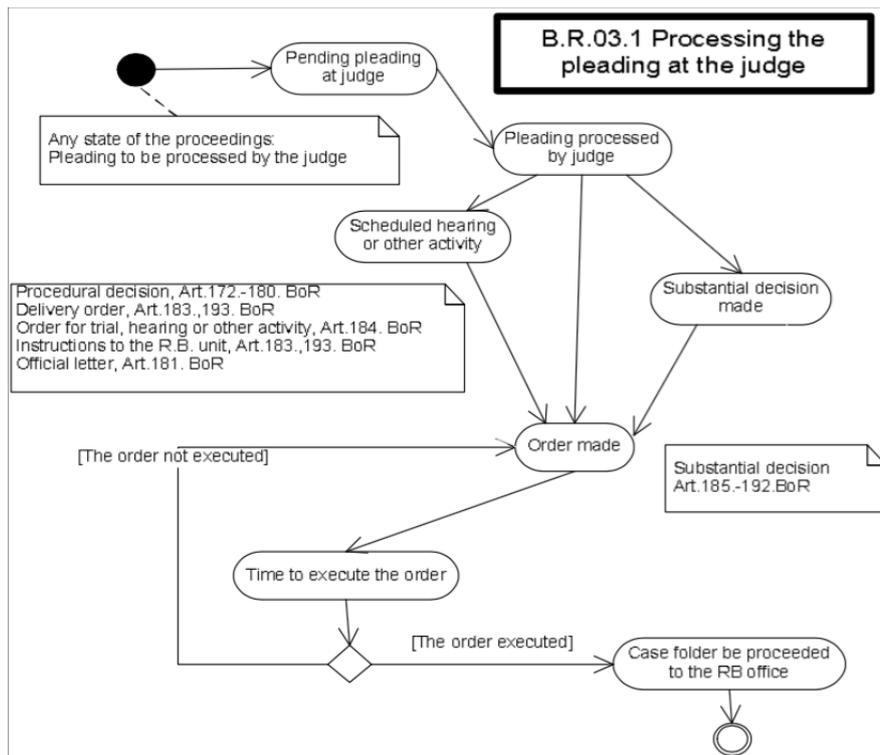
Quelle: Soltana et al. (2014, S. 8)

Abbildung 2.10.: Beispiel eines Aktivitätendiagramms für die Absetzung von Pendlerkosten

Tools durchzuführen. Auch Bork und Fill (2014, S. 7) bestätigen den Nutzen von UML Profiles: "Besides the already defined set of diagrams by the OMG, modelers have the possibility to create UML profiles. These UML profiles enable the UML to be applicable in new and emergent domains."

Strahonja (2006) und van Engers et al. (2001) benutzen zur Rechtsmodellierung UML state machine Diagramme. UML state machine Diagramme sind Zustandsdiagramme, die alle möglichen Zustände eines Objekts im Verlauf anzeigen. Zusätzlich ist auch die Ursache der Zustandsänderungen ersichtlich. Die UML-Zustandsdiagramme sind gerichtete Graphen, in denen Knoten Zustände und Verbindungen zwischen den Knoten die Zustandsübergänge bezeichnen. Ihr Zweck ist das "modeling discrete behavior through finite state transition system" (OMG, 2009, S. 541). Die UML state machine Diagramme lassen sich in behavioral(Verhalten) und protocol(Protokoll) state machines unterscheiden. Verhaltens-Zustandsmaschinen können verwendet werden, um das Verhalten verschiedener Modellelemente wie beispielsweise Klasseninstanzen

festzulegen (OMG, 2009). Protokoll-Zustandsmaschinen werden verwendet, um Nutzungsprotokolle zu erstellen. Zum einem ist somit durch die verschiedenen Formen der Zustandsdiagramme das Verhalten eines Systemteils nachvollziehbar, zum anderen können Zustandsmaschinen auch dazu verwendet werden, um ein Nutzungsprotokoll zu erzeugen.



Quelle: Strahonja (2006, S. 2)

Abbildung 2.11.: UML State Machine Diagramm für Gerichtsprozesse

Strahonja (2006) modellieren Rechtstexte mit UML state machine Diagramme, um einerseits das domänenspezifische Wissen in Modelle zusammenzufassen und andererseits "[...] by using commonly agreed and understandable modeling concepts of the Unified Modeling Language (UML)[.]" (S.1). Außerdem verfolgen sie die Absicht Court Management Systeme zu ermöglichen. Dazu müssen als Voraussetzung Gesetzestexte als Systemmodelle vorliegen. Ein beispielhaftes UML State Machinge Diagramm für einen Geschäftsprozess ist in Abbildung 2.11

abgebildet, wo in den Elementen außerhalb des Workflows die entsprechenden Verweise auf relevante Gesetze und Gesetzesartikel verwiesen wird.

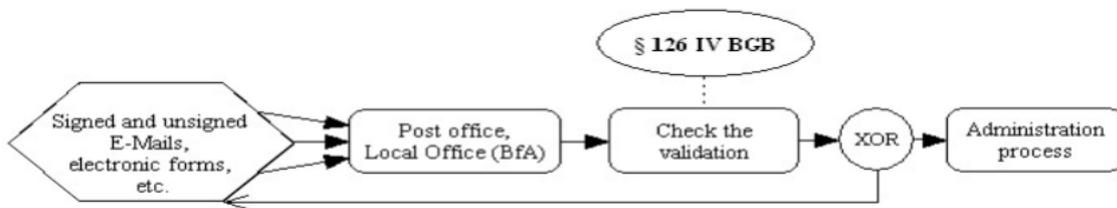
UML bietet zwar einen allgemeinen Einstieg in die Rechtsmodellierung, zumal es vor allem in der Softwareentwicklung weit verbreitet und populär ist, bringt aber auch einige Nachteile mit sich. Es bedarf genauester Überlegungen und Definitionen, wie UML für Gesetzestexte eingesetzt werden kann. Einen Standard für das Recht gibt es nämlich nicht, da UML typischerweise in der Softwareentwicklung verwendet wird.

Geschäftsprozessmodelle

Besonders im Bereich der Geschäftsprozessmodellierung ist die Relevanz der Gesetzgebung zu betonen (Olbrich, 2007; Awad und Weske, 2010). Dabei werden oft juristische Experten bei der Planung und Durchführung der Prozesse zu Rat gezogen, um deren gesetzeskonformen Ablauf zu garantieren und gegen keine externen Reglementierungen zu verstoßen. Eine Umfrage über die Anfälligkeit von Organisationsformen aufgrund von externen Änderungen bestätigt zudem, dass es sich bei einem der Gründe um den Einfluss der Reglementierungen handelt (Scheer, Kruppke und Heib, 2003). Auch der Aspekt der Einhaltung von Compliance Regelungen bei Geschäftsprozessen ist zu beachten (siehe auch Kapitel 2.2). Compliance Regelungen schließen oft gesetzliche Regulierungen ein.

Der Vorschlag von Alpar und Olbrich (2005) war daher die gesetzlichen Bestimmungen als Informations- und Organisationssymbole mit in die ereignisgesteuerte Prozesskette (EPK) zu nehmen, da auf diese Weise in erster Linie eine Dokumentation der reglementierten Abläufe sichergestellt ist.

Zudem ermöglichen die zusätzlichen Modellierungselemente nicht nur Validierungen, Simulationen und Aufwandsabschätzungen sondern auch Überprüfungen auf Automatisierungsmöglichkeiten und die Auswirkungen von Gesetzesänderungen. Wie so eine EPK aussehen könnte, ist in Abbildung 2.12



Quelle: Alpar und Olbrich (2005, S. 111)

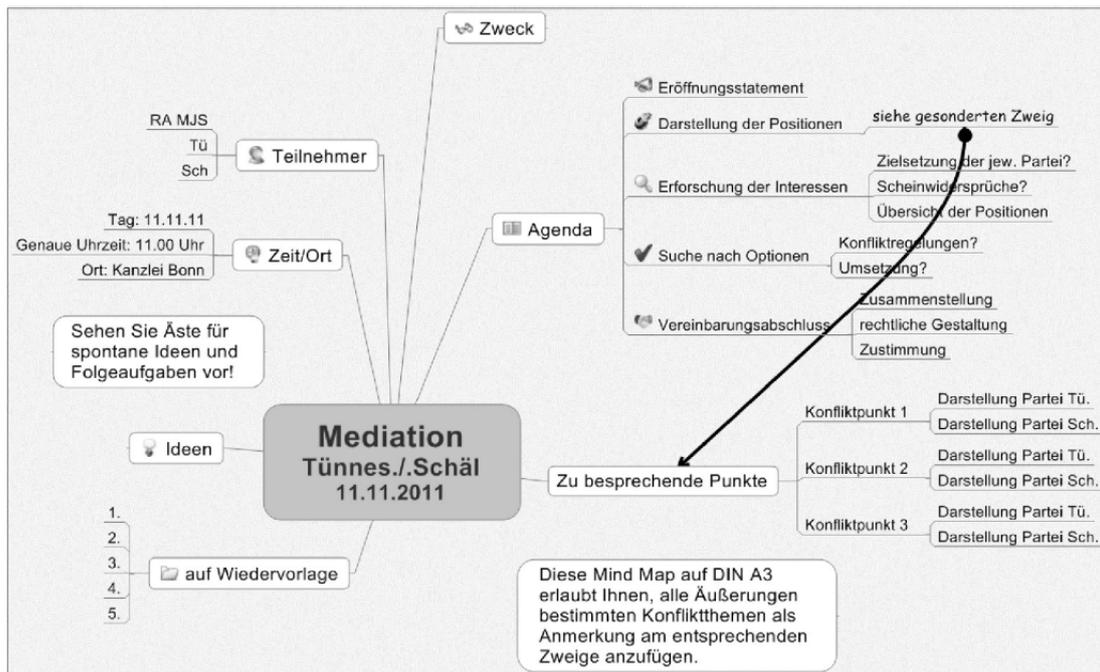
Abbildung 2.12.: Beispiel-EPK mit Elementerweiterungen für gesetzliche Bestimmungen

zu sehen, wo die Validierung von elektronisch signierten und unsignierten Nachrichten mit Hilfe von § 126 IV BGB abgebildet ist.

Allerdings muss hinzugefügt werden, dass es sich bei der vorgeschlagenen Methode um eine Visualisierung von rechtlichen Rahmenbedingungen handelt, welche nur einen Informationscharakter im Bezug auf die relevanten Paragraphen des Prozesses besitzen. In diesem Sinne wird das Recht nicht modelliert, sondern nur darauf verwiesen.

Mind Mapping

Sauerwald (2007) stellt Mind Mapping als Visualisierungsmethode in der juristischen Praxis vor. Dabei sollen diese der Vereinfachung der Komplexität beispielsweise bei Gesprächsvorbereitungen, -führungen und nachbereitungen oder bei Mandatsplanungen dienen. Da das Mind Mapping zudem einfach und schnell erlernbar ist, eignet sich die Technik für diesen Einsatzzweck besonders. Mind Maps können jedoch ausschließlich erstellt werden, wenn der Ersteller ausreichend Kenntnisse über die Rechtslage hat. Auch hier ist wie im Kapitel zur Visualisierung in Geschäftsprozessmodellen festzustellen, dass Mind Mapping nur einen Informationscharakter besitzt und zudem nur für spezifische Situationen einsetzbar ist. In Abbildung 2.13 stellt Sauerwald (2007) ein Mindmap zu einem Mandantengespräch dar.



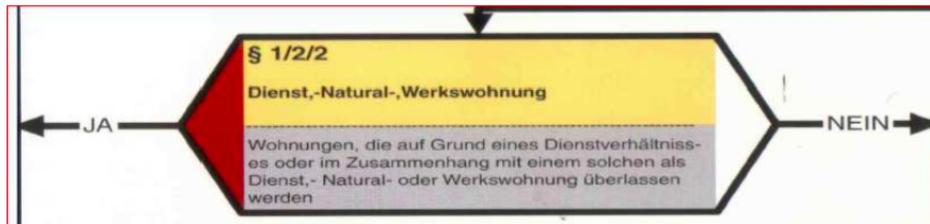
Quelle: Sauerwald (2007, S. 481)

Abbildung 2.13.: Beispiel eines Mindmaps zur Mediation in einem Mandantengespräch

Eigens entwickelte Modellierungsansätze

Kahlig (2011; 2013) entwickelt für Fragen bezüglich des Miet- und Immobilienrechts eine Modellierungsmethode, die es ermöglicht rechtliche Fragen zu klären. Dabei beinhaltet die Modellierungsmethode neben einem Start und Ende, Workflow-Elemente, logische Verbindungselemente, Aktionen und Ja-oder-Nein-Entscheidungselemente.

Dieses Schema beruht auf der Idee, dass es zu jeder Frage nur die Antwortmöglichkeiten "Ja" oder "Nein" gibt. Die Aneinanderreihung von diesen geschlossenen Fragen soll somit einen Anwendungsfall abdecken, indem die Abarbeitung aller relevanter Fragen zur Klärung von rechtlichen Fragen führen.



Quelle: Kahlig und Kahlig (2013, S. 10)

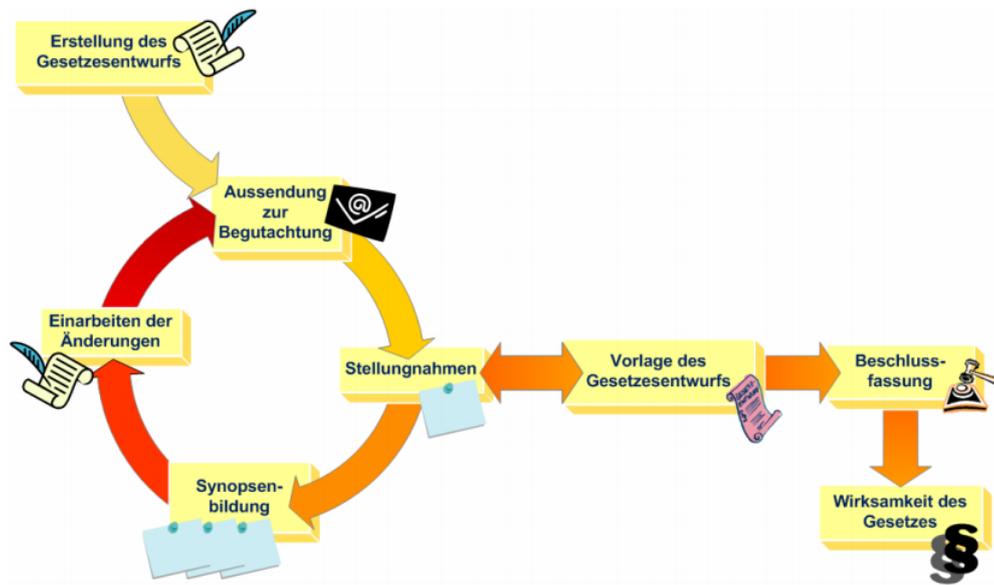
Abbildung 2.14.: Beispiel eines Ja-oder-Nein-Entscheidungselement

2.4.3. Softwareprojekte im deutschsprachigen Raum

In diesem Kapitel sollen zwei Softwareprojekte aus dem deutschsprachigen Raum vorgestellt werden. Dazu zählt zum einen das österreichische Lexwas Projekt und zum anderen das deutsche Digital-Humanities-Projekt ARGUMENTUM.

Lexwas Projekt

Mit dem Projekt Lexwas, welches eine Zusammenarbeit zwischen Bundesministerium für soziale Sicherheit und Generationen und der Technischen Universität Wien ist, soll die Rechtsetzung elektronisch unterstützt werden. Dabei soll "der gesamte Prozess der Gesetzwerdung abgebildet werden (Entwurf, Iteration, Inkrafttreten, Revision)" (INSO, 2008). Dieser Prozess ist in Abbildung 2.15 genauer dargestellt. Schefbeck (2007) erläutert in seinem Beitrag zu den Klagenfurter Legistikgesprächen 2007 die Bedeutung der E-Partizipation in Zusammenhang des Projekts Lexwas, welche in erster Linie einen legistischen XML-Editor bereitstellen soll. Im Jahre 2004 ergab sich dem anschließend das Model LEXWAS-Begutachtung, "[...] welche die strukturierte Abgabe von Stellungnahmen zu einem zuvor in XML strukturierten Rechtsakt erlaubt" (Schefbeck, 2007)[S. 17]. Dieses XML mit den Eingaben des Benutzers auf einer Weboberfläche wird als Entwurf in einer Word-Datei zusammengefasst und kann dann zur Begutachtung freigegeben werden.



Quelle: INSO (2008)

Abbildung 2.15.: Lebenszyklus des Gesetzes

Digital-Humanities-Projekt ARGUMENTUM

In der juristischen Praxis kann man als eines der komplexesten Aufgaben die Findung und Strukturierung von Argumentationsketten in Einzelfällen bezeichnen. Zu dieser Thematik wird das Konsortialforschungsprojekt ARGUMENTUM in der Bundesrepublik Deutschland durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert. Ziel des Projekts ist es, eine Software zur "automatisierten Identifikation und Analyse von Argumentationsstrukturen in Gerichtsurteilen [...]" (Houy et al., 2012, S. 2) zur Verfügung zu stellen. Dabei soll die Software ARGUMENTUM Methoden wie Information Retrieval und NLP nutzen und Juristen eine Möglichkeit bieten die anfänglich beschriebene Hürde mit Hilfe eines geeigneten Werkzeugs zu überwinden.

Grundlage zur Entwicklung der Software sind Gerichtsentscheidungen, aus denen Argumentationen identifiziert, extrahiert und in kleinere Einheiten wie Sätze und Tokens zerlegt werden. Diese ermöglichen es dann nach bestimmten Argumenten oder Argumentationsmustern zu suchen. Bekannte NLP Methoden

The screenshot displays the ARGUMENTUM software interface. On the left, a 'Gliederungsübersicht' (Table of Contents) shows a hierarchical structure of a legal case, starting with '1bvr035705' and branching into sections B, C, and D, with further sub-sections like I, II, 1, 2, a), b), aa), bb), (1), (2), (a), (b), (c).

The main text area shows a legal decision text with several paragraphs. The first paragraph discusses the scope of the term 'imminent danger' (unmittelbar drohende Gefahrenzustände). The second paragraph, highlighted in yellow, states: 'Auch der Sinn und Zweck des Art. 35 Abs. 2 Satz 2 GG, den Bund zu einer effektiven Hilfeleistung im Aufgabenbereich der Länder zu befähigen, spricht dafür, beide Katastrophenursachen in zeitlicher Hinsicht gleich zu behandeln und in beiden Fällen mit der Hilfeleistung nicht abzuwarten, bis die zum Schadensereignis führende Gefahrenentwicklung ihren Abschluss gefunden hat.'

Below this, two numbered paragraphs (103 and 104) discuss the deployment of federal forces (Streitkräfte) in the event of a natural disaster, referencing Art. 35 Abs. 2 Satz 2 GG and the Basic Law (Grundgesetz).

On the right, there is a sidebar titled 'enthaltene Argumentationsmuster' (Contained Argumentation Patterns). It lists several patterns with checkboxes: 'Auslegung nach Sinn und Zweck', 'Historische Auslegung', 'Güterabwägung', 'Systematische Auslegung', and 'Auslegung nach Wortlaut'. Below this, there is a section 'Argumentationsmuster-Info' which shows 'Zugeordnete Argumentationsmuster für Art. 35 Abs. 2 Satz 2 GG (Rn. 102)' and 'Auslegung nach Sinn und Zweck ++'.

Quelle: Houy et al. (2015)

Abbildung 2.16.: "Detaildarstellung mit annotierten und markierten Argumentationsmustern"

wie z. B. Named-Entity-Recognition, Part-of-Speech-Tagging, Lemmatisation oder Stemming kommen dabei zum Einsatz (Houy et al., 2015).

Diese Software ermöglicht es somit die tägliche Arbeit von Forschern und Juristen zu erleichtern, indem es eine elektronische Suchfunktion in vorhandene Argumentationsketten zur Verfügung stellt. Gleichzeitig wird der Suchradius auf elektronische Sachverhalte ausgedehnt und beschränkt sich nicht nur auf gedruckte Quellen. Zudem ist es möglich den Aufbau der Argumentationsstruktur anhand einer großen Datenbasis zu analysieren und Schlüsse zu ziehen. Diese sogar in Argumentationsmustern zu clustern und den Juristen als erfolgreiche Argumentationskette bei verwandten Themen vorzuschlagen (Houy et al., 2015). Eine Detailansicht der Bedienoberfläche der Software ist in Abbildung 2.16 mit markiertem Argumentationsmuster zu sehen.

2.4.4. Anwendungen in anderen Ländern

Neben den vorhandenen Modellierungsansätzen und Softwareprojekten im deutschsprachigen Raum sollen in diesem Artikel kurze Einblicke in Anwendungen zum Information Retrieval im juristischen Kontext in anderen Ländern gegeben werden. Als Beispiele dienen hier die Niederlande, Tasmanien und die USA.

Niederlande

Die niederländische Verwaltung für Steuern und Recht beteiligt sich am Forschungsprojekt POWER ("Program for an Ontology based Working Environment for Regulations and legislation"), welche ihren Schwerpunkt in der Übersetzung von Gesetzen in Prozesse setzt (Van Gog und Van Engers, 2001). Dazu wird zunächst versucht das Recht zu formalisieren, um diese Formalismen dann in Modellelemente zu übersetzen und schließlich zu modellieren. Van Gog und Van Engers (2001) entwickeln dazu das Tool OPAL (Object-oriented Parsing and Analysis of Legislation), um Gesetzestexte automatisch zu modellieren, indem sie versucht haben "to identify the object oriented concepts in the legislation written in natural language" (S. 562).

Tasmanien

In Tasmanien existiert ein zum österreichischen ähnliches Rechtsinformationssystem, welches allerdings die Möglichkeit bietet automatisiert konsolidierte Fassungen der Gesetze zu erstellen (Reimer, 2008). Dabei kann der Legist Rechtstexte aus der Datenbank abfragen und diese in einem beliebigen Textverarbeitungsprogramm editieren. Aus den Änderungen wird ein sogenanntes Change Description Document (CDD) erstellt, welches die Novellierungsanordnungen enthält. Um eine konsolidierte Fassung zu erhalten, wird das CDD mit Hilfe eines Amendment Wording Generator (AWG) auf den Rechtstext angewendet. Somit bedarf es im Gegensatz zum österreichischen Ansatz nicht extra geschultes Personal, welche die Anordnungen in bestehende Gesetze einarbeitet

(Reimer, 2008).

USA

Ein interdisziplinäres Team der Stanford Law School entwickelte eine Online-Plattform, die Visual Law Library, um den Umgang mit Gesetzen für Jurastudenten zu vereinfachen und mit Hilfe von Visualisierung zu unterstützen. Die offen gestaltete Plattform (<http://www.legaltechdesign.com>) ist als Bibliothek gedacht und unterstützt die Partizipation zwischen der Community, sodass jeder angehalten wird einen Beitrag zu teilen.

2.4.5. Diskussion

Um einen besseren Überblick über die vorgestellten Ansätze zu erhalten, werden diese im Nachfolgenden anhand von bestimmten Kriterien verglichen und bewertet. Diese Kriterien lauten:

1. Zielgruppe
2. Modellierung der Struktur
3. Modellierung der Semantik
4. Expertenwissen notwendig?
5. Benutzerfreundlichkeit (hoch, mittel, niedrig)
6. Kognitive Belastung (hoch, mittel, niedrig).

Dabei beschreibt die Zielgruppe die Benutzer der Applikation bzw. des Forschungsprojekts, für welche drei Gruppen nach Fill und Grieb (2017) vorgeschlagen werden: Rechtsexperten (RE), Fachexperten (FE) und rechtliche Laien (RL). Die Punkte 2 und 3 stellen die Frage, ob eine Modellierung für die Struktur (2) und/oder eine für die Semantik (3) existiert. Mit Punkt 4 soll festgestellt werden, ob ein Experte bei der Bedienung notwendig ist oder das System auch von Laien

2. Grundlagen und verwandte Arbeiten

Ansatz	Zielgruppe	Syntax	Semantik	Expertenwissen	Benutzerfreundlichkeit	Kog. Belastung
UML	RE, FE, RL	ja	ja	ja	mittel	hoch
Mind Mapping	RE, FE	nein	ja	ja	hoch	niedrig
Geschäftsprozessmodellierung	RE, FE, RL	nein	nein	ja	mittel	niedrig
Kahlig Ansatz	RE, FE, RL	nein	ja	ja	hoch	niedrig
NLP	RE, FE	teilweise	teilweise	nein	hoch	mittel

Tabelle 2.2.: Vergleich der Modellierungsansätze
(FE: Fachexperte, RE: Rechtsexperte, RL: Rechtlicher Laie)

Software	Zielgruppe	Syntax	Semantik	Expertenwissen	Benutzerfreundlichkeit	Kog. Belastung
Lexwas TU Wien	RE, FE	nein	nein	ja	mittel	hoch
ARGUMENTUM	RE, FE	nein	ja	ja	hoch	mittel

Tabelle 2.3.: Vergleich der Editoren
(FE: Fachexperte, RE: Rechtsexperte, RL: Rechtlicher Laie)

genutzt werden kann. Für die Punkte 2 bis 4 wird die Auswahl auf *ja*, *nein* oder *teilweise* beschränkt. Das Einstufen der Benutzerfreundlichkeit ist von *hoch* bis *niedrig* möglich, ebenso die kognitive Belastung des Users.

Dieser Vergleich findet sowohl nach den gewählten Ansätzen, den Autoren, den Editoren als auch nach den länderspezifischen Ansätzen statt und ist in Tabelle 2.2, 2.5, 2.3 und 2.4 zusammengefasst.

Kein vorgestellter Ansatz/ Autor erfüllt bisher alle Bewertungskriterien, welche für die Motivation dieser Arbeit dienen. Diese Tatsache wirft die Frage nach einem angemessenen Modellierungsansatz auf. Dieser sollte sowohl die Struktur der Gesetze abbilden, als auch semantische Analysen ermöglichen. Zudem sollte idealerweise eine große Zielgruppe angesprochen werden, sodass kein Experte beim Benutzen der Modellierungsmethode von Nöten ist. Eine hohe Benutzerfreundlichkeit und eine niedrige kognitive Belastung sind dabei Vo-

Land	Zielgruppe	Syntax	Semantik	Expertenwissen	Benutzerfreundlichkeit	Kog. Belastung
Niederlande	RE, FE, RL	ja	teilweise	teilweise	mittel	mittel
Tasmanien	RE, FE, RL	nein	nein	ja	mittel	hoch
USA	RL	nein	ja	nein	mittel	mittel

Tabelle 2.4.: Vergleich der länderspezifischen Ansätze
(FE: Fachexperte, RE: Rechtsexperte, RL: Rechtlicher Laie)

2.4. Information Retrieval im juristischen Kontext

Autor	Zielgruppe	Syntax	Semantik	Expertenwissen	Benutzerfreundlichkeit	Kog. Belastung
Alpar und Olbrich (2005)	RL	nein	teilweise	ja	mittel	hoch
Engesser (2014)	RE, FE	ja	ja	ja	mittel	hoch
van Engers et al. (2001)	RE, FE, RL	nein	ja	ja	mittel	niedrig
Hahn, Mielke und Wolff (2013)	RL	nein	ja	nein	hoch	niedrig
Kahlig (2011) Kahlig und Kahlig (2013)	RE, FE, RL	nein	ja	ja	mittel	niedrig
Merkel, Schweighofer und Winiwarter (1994)	RE, FE	ja	ja	ja	mittel	mittel
Sauerwald (2007)	RE, FE	nein	teilweise	ja	mittel	niedrig
Schoormann und Knackstedt (2016)	RL	nein	ja	nein	hoch	niedrig
Schweighofer (1999)	RE, FE	ja	ja	ja	mittel	mittel
Schweighofer und Winiwarter (1993)	RE, FE	ja	ja	ja	mittel	mittel
Schweighofer, Winiwarter und Merkl (1995)	RE, FE	ja	ja	ja	mittel	mittel
Soltana et al. (2014)	RE, FE, RL	nein	ja	ja	mittel	niedrig
Strahonja (2006)	RE, FE, RL	nein	ja	ja	mittel	niedrig
Tobler und Beglinger (2010)	RL	nein	ja	nein	hoch	niedrig
Van Gog und Van Engers (2001)	RE, FE	ja	ja	ja	hoch	mittel
Walter (2009)	RE, FE	nein	ja	ja	hoch	mittel
Wyner et al. (2010)	RE, FE	nein	ja	ja	hoch	mittel

Tabelle 2.5.: Vergleich der Ansätze nach Autoren
(FE: Fachexperte, RE: Rechtsexperte, RL: Rechtlicher Laie)

raussetzungen. Auch die länderspezifischen Ansätze wie den tasmanischen Ansatz mit automatisierten konsolidierten Fassungen könnte man mit Hilfe einer Modellierungsmethode und der Protokollierung der Änderungen mit Hilfe geeigneter technischer Mittel umsetzen.

Den Ergebnissen entsprechend wird zum Ziel gesetzt zunächst die Struktur der Gesetze mit Hilfe einer Modellierungsmethode darzustellen und sich dann anschließend der Semantik zu widmen. Da zudem Verweisungen zwischen und innerhalb der Gesetze bis zur Satzebene existieren, wird als erstes Ziel die Erkennung der Sätze im Gesetzestext gesetzt. Dieser Schritt soll zusätzlich die Notwendigkeit der Markierung von Sätzen schon vom Legisten unterstreichen. Schweighofer, Winiwarter und Merkl (1995) bestätigen die Ansicht, dass "[a]dvanced use of information technology in the legal field requires some formalisation of the legal data[.]" (S. 1), sodass eine Darstellung der Syntax von Gesetzestexten mit dieser Arbeit vorangetrieben und umgesetzt wird.

2.5. Konzepte zur Satzmustererkennung

Durch das vorgegebene Schema für Gesetzestexte aus dem Bundeskanzleramt (1990) und dem daraus hergeleiteten Aufbau von Gesetzestexten (siehe Abbildung 2.2) ist abzuleiten, dass die kleinste Gliederungseinheit für Rechtstexte die des Satzes bzw. Teilsatzes ist. Diese Gliederungseinheit kann sogar bis zur geringst möglichen Stufe - des alphabetischen Buchstaben - herunter gebrochen werden. Auch für die zahlreichen Verweisungen in den Texten, wie in Kapitel 2.1.2 beispielhaft gezeigt, die dafür sorgen, dass die Übersichtlichkeit und Verständlichkeit von Gesetzen leidet, soll eine Möglichkeit geschaffen werden, diese für den Leser zugänglich zu machen. Eine Möglichkeit wäre diese Referenzen im Fließtext zu erkennen und eine Verknüpfung zur entsprechenden Textstelle zu erzeugen. Verweisungen in österreichischen Gesetzen liegen allerdings bis zur Ebene des Satzes bzw. Teilsatzes vor. Dieser Aspekt macht eine Analyse auf Satzebene nötig, um eine Basis für die automatisierte Erkennung der Referenzen zu schaffen.

Wie Palmer und Hearst (1997) feststellt, ist die Punktierung als typischer Marker für das Ende eines Satzes häufig nicht eindeutig. Die Gründe dafür liegen in der Verwendung von Abkürzungen, aber auch in Besonderheiten der Textkategorie, kurz gesagt in jeglicher Form von Doppeldeutigkeit, welche ohne spezielle Regeln oder Betrachtung des Kontextes nicht aufgehoben werden kann. In den folgenden Kapiteln werden daher einige grundlegende Konzepte zur Satzmustererkennung beschrieben und evaluiert, welche am geeignetesten für die vorliegende Problematik sind.

2.5.1. Direkter Ansatz

Ein naiver Ansatz kann mit Hilfe von hartkodierten Routinen zur Satzerkennung verfolgt werden. Dabei geht bei Erstellung keine Form der Modellierung der Wissensbasis z.B. durch Erstellung von abstraktem Regelwerk zur Satzerkennung voraus. Es werden Abfragen direkt als Schleifen und vordefinierten Konstrukten angefertigt, die prozedural überprüfen, ob ein Satz vorliegt. Möglich ist zudem

die Implementierung eines Abkürzungsverzeichnisses, um die Genauigkeit der Satzerkennung zu steigern. Eine beispielhafte Routine für eine Satzerkennung wurde dafür in Pseudocode erstellt und wird in Abbildung 2.17 angezeigt.

Wie Walker et al. (2001) feststellt, ist es mit diesem Ansatz relativ schnell möglich eine Satzerkennung umzusetzen. Allerdings stellt die Wartung und Erweiterung ein massives Problem dar. So ist es architekturbedingt nicht möglich die Implementierung ohne erheblichen Aufwand an Änderungen der Sprache anzupassen oder gar

```

1  for(wort in woerter) {
2      if (istWort(nachfolgerVon(wort))) {
3          then marker(wort) = keinSatzende;
4      } else if(wort in abkuerzungsverzeichnis) {
5          then marker(wort) = abkuerzung;
6      } else if(nachfolgerVon(wort) == ".") {
7          then marker(wort) = satzende;
8      } else {
9          //to be continued
10     }
11 }
```

auf eine weitere Sprache auszuweiten. **Abbildung 2.17.:** Beispiel einer hartkodierten Routine als Pseudocode

2.5.2. Regelbasierter Ansatz

Um den Problemen des naiven Ansatzes gerecht zu werden, ist es im Allgemeinen notwendig Systeme zur Satzerkennung mit einem speziellem Regelwerk auszurüsten, welches den Besonderheiten der zu verarbeitenden Dokumente (wie z.B. Gesetzestexten) entsprechend Rechnung trägt. Walker et al. (2001) führen an, dass die deklarative Speicherung der linguistischen Wissensbasis eine Änderung und Erweiterung der Satzerkennung um weitere Sprachen stark vereinfacht. Es wird nur noch die Änderung bzw. Erstellung einer Grammatik verlangt, im Gegensatz zur Neuimplementierung sämtlicher beteiligter Routinen, wie es beim naiven Ansatz der Fall ist.

Eine Regel des regelbasierten Ansatzes kann folgendermaßen aussehen: (.*[?!.]

Dieser reguläre Ausdruck splittet einen vorliegenden Text nach jedem Fragezeichen, Ausrufezeichen oder Punkt. Der Text *"Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet."* wird dementsprechend in folgende Teile geteilt:

- "Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua."
- " At vero eos et accusam et justo duo dolores et ea rebum."
- " Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet."

Doch auch die Erstellung einer Grammatik geht offensichtlich mit einem erheblichem Aufwand einher und der Ansatz ist nicht generisch einsetzbar. Eine Alternative stellen Methoden dar, welche eine Satzbestimmung anhand von statistischen Daten durchführen.

2.5.3. Heuristische Methoden

Heuristische Methoden zur Satzerkennung bieten große Flexibilität, da es das Ziel dieser Vorgehensweisen ist, die Grenzen eines Satzes eindeutig bestimmen zu können, ohne, dass dafür auf spezielle Regeln zurück gegriffen werden muss. Reynar und Ratnaparkhi (1997) stellen eine Herangehensweise vor, welche auf dem Maximum-Entropie Model basiert. Dabei wird für einen gegebenen Kontext eines Tokens die Wahrscheinlichkeit einer Satzgrenze anhand der Wahrscheinlichkeitsverteilung der maximalen Entropie unter bestimmten Einschränkungen geschätzt. Als Beispiel für einer ihrer Methoden bringen Reynar und Ratnaparkhi (1997, S. 17) folgende sechs Charakteristika für einen Satz an:

- den Prefix

- den Suffix
- ob entweder Prefix oder Suffix eine Abkürzung ist
- das Wort links vom betroffenen Wort
- das Wort rechts vom betroffenen Wort
- ob das Wort links oder rechts vom betroffenen Wort eine Abkürzung ist.

Für den Beispielsatz "(1) ANLP Corp. chairman Dr. Smith resigned." lauten demnach die Zuordnungen: "[...]PreviousWord = ANLP, FollowingWord = chairman, Prefix = Corp, Suffix = NULL, PrefixFeature = InducedAbbreviation[.]" Reynar und Ratnaparkhi (1997, S. 17).

Eine Abkürzungsliste wird automatisch von Trainingsdaten erstellt. Die identifizierten Zuordnungen werden anschließend sogenannten Features zugeordnet, mit Hilfe derer dann eine Wahrscheinlichkeit nach dem Maximum-Entropie Model ausgerechnet wird. Dieser bestimmt die Wahrscheinlichkeit, ob es sich bei dem erkannten Satzzeichen um ein Satzende handelt oder nicht.

Auch bei Palmer und Hearst (1997) trifft das System jeweils einfache Annahmen über die Token vor und nach einem Satzzeichen. Diese Annahmen dienen einem selbstlernendem Algorithmus als Argumente, welcher dann eine entsprechende Klassifizierung des Satzzeichens vornimmt.

2.5.4. Stanford Log-linear Part-Of-Speech Tagger

Der Log-linear Part-Of-Speech Tagger der Universität Stanford ist eine Software, welche es für eine Vielzahl von Sprachen ermöglicht nicht nur Sätze zu erkennen, sondern auch die jeweilige Wortart zu den entsprechenden Token zuzuordnen. In der von Toutanova et al. (2003) verfassten theoretischen Ausarbeitung, auf welcher die Java Implementierung basiert, wird ein System beschrieben, das die im vorherigen Kapitel beschriebene heuristische Methode der maximalen Entropie erweitert und dabei sehr gute Ergebnisse liefert. Für die Satzerkennung kann man mit Hilfe des Part-Of-Speech Taggers Regeln definieren, welche

alle möglichen Abfolgen von Wortarten abdecken, um einen Satz zu erkennen. Eine mögliche Regel wäre: */NN /VFIN /APPR /ART /NN\$.*, wobei */NN* für normales Nomen, */VFIN* für finites Verb, */APPR* für Präposition, */ART* für bestimmter oder unbestimmter Artikel steht (Stuttgart, 1995) und *\$.* einen Satzpunkt markiert.

Dabei ist der Stanford Tagger jedoch vor allem auf die englische Sprache spezialisiert und nur bedingt für österreichische Rechtstexte geeignet. Vor allem Abkürzungen der deutschen Sprache stellen Hindernisse für die korrekte Markierung und Erkennung von Satzgrenzen dar.

2.5.5. Weitere Ansätze

Für die Satzerkennung, welche man nach einer Klassifizierung anhand von statistischen Daten vornimmt, sowie breiter gefasst für die automatische Wortart-Zuweisung, existieren zahlreiche weitere Ansätze. Zu nennen sind dabei vor allem gut funktionierende Ansätze, welche auf verschiedenen Methoden aus dem Feld des Machine-Learning zurückgreifen. Eine ausführliche Übersicht findet sich dabei in Manning und Schütze (1999).

2.5.6. Diskussion

Der Anwendungsfall, der in der vorliegenden Arbeit diskutiert wird, grenzt sich von den Anwendungsfällen für welche statistische Methoden relevant sind deutlich ab. Die Anforderung einer hohen Flexibilität, beliebige Texte einer Sprache korrekt verarbeiten zu können, ist für diese Arbeit nicht in diesem Maße relevant.

Es handelt sich bei einem österreichischem Gesetzestext viel mehr um einen Text, welcher nach strikten Regeln verfasst wurde. Es werden relativ wenig Regeln benötigt, um eine vollständig korrekte Satzerkennung durchzuführen. Nach eingehender Analyse ist darüber hinaus die Anzahl der vorkommenden Abkürzungen, zumindest ihre Muster jedoch, stark begrenzt. Der Einsatz eines statistischen Verfahrens impliziert zudem auch immer die Tolerierung eines

gewissen Prozentsatzes an falsch kategorisierten Worten. Dies ist für die Verarbeitung von Gesetzestexten nicht akzeptabel. Für die Umsetzung einer Satzerkennung wird sich daher für einen regelbasierten Ansatz mit Hilfe von Regular Expressions entschieden.

3. Implementierung einer Satzmustererkennung für Gesetzestexte

Wie in Kapitel 2.5 bereits erläutert, ergibt die Analyse des Aufbaus und der Syntax von Gesetzestexten eine Darstellung bis zur Gliederungseinheit des Satzes in Modellen. Zu diesem Zwecke wird eine Satzmustererkennung für Gesetzestexte implementiert und soll in diesem Kapitel im Detail vorgestellt werden.

3.1. Vorgehen

Für die in dieser Arbeit umgesetzte Implementierung sind die zur Satzerkennung genutzten Regeln in Form regulärer Ausdrücke bzw. *Regular Expressions* vom Benutzer zu definieren und können für spezifische Fälle eingesetzt, sowie gegebenenfalls erweitert werden. Letzteres bietet demnach vor allem die Möglichkeit spezifischer Anpassungen an österreichische Gesetzestexte.

Zur Ermittlung der notwendigen regulären Ausdrücke für das österreichische Recht wird für die Analyse ein Beispielgesetz herangezogen. Dabei handelt es sich um das Mietrechtsgesetz (MRG). Anhand des MRGs wird zunächst eine manuelle Analyse durchgeführt und Satzmuster identifiziert, welche dann mit Hilfe von regulären Ausdrücken formal abgebildet werden, um auf dieser Basis eine automatisierte Satzmustererkennung zu entwickeln.

3.2. Manuelle Analyse

Das Ziel der manuellen Analyse ist die Identifizierung von Satzmustern, um die Sätze in einem Gesetzestext über ein automatisiertes Verfahren herauszufinden. Dazu bedarf es zunächst einer ausführlichen Analyse, die manuell durchgeführt wird. Das menschliche Gehirn erkennt schnell in einem vorliegenden Text die Sätze, so dass der erste Schritt der manuellen Analyse eine Erkennung der Sätze im Fließtext voraussetzt. In Abbildung 3.1 ist dementsprechend ein Satz markiert.

Um eine maschinenlesbare Form zu bearbeiten, sind vom RIS die Gesetzestexte im XML-Format heruntergeladen. Alle Sätze im RIS-XML des MRG sind dann mit dem XML-Tag `<satz>` manuell markiert, womit gleichzeitig auch Satzmuster identifiziert werden konnten. Ein Ausschnitt eines annotierten RIS-XML in Abbildung 3.2 zeigt das Ergebnis der manuellen Analyse, als Beispiel dient hier wieder §2 Abs. 1 Satz 1 bis 3 MRG.

Die manuelle Analyse des MRG,

```
[...] § 2. (1) Hauptmiete liegt vor, wenn der Mietvertrag mit dem Eigentümer oder dem dinglich oder obligatorisch berechtigten Fruchtnießer der Liegenschaft oder mit dem Mieter oder Pächter eines ganzen Hauses geschlossen wird. Steht der Mietgegenstand im Wohnungseigentum, so wird Hauptmiete durch den Mietvertrag mit dem Wohnungseigentümer begründet. Wenn am Mietgegenstand Wohnungseigentum erst begründet werden soll, geschlossenen Mietvertrag Hauptmiete mit dem Eigentümer oder den Eigentümern der Liegenschaft zustande, doch geht mit der Begründung von Wohnungseigentum am Mietgegenstand die Rechtsstellung des Vermieters auf den Wohnungseigentümer über. [...]
```

Quelle: eigene Darstellung

Abbildung 3.1.: Schritt 1: Durchführung einer manuellen Analyse (§2 Abs. 1 Satz 1 bis 3 MRG)

```
[...] <absatz typ="abs" ct="text" halign="j">  
<gldsym>§ 2.</gldsym> (1)  
<satz>Hauptmiete liegt vor, wenn der Mietvertrag mit dem Eigentümer oder dem dinglich oder obligatorisch berechtigten Fruchtnießer der Liegenschaft oder mit dem Mieter oder Pächter eines ganzen Hauses geschlossen wird.</satz>  
<satz>Steht der Mietgegenstand im Wohnungseigentum, so wird Hauptmiete durch den Mietvertrag mit dem Wohnungseigentümer begründet.</satz>  
<satz>Wenn am Mietgegenstand Wohnungseigentum erst begründet werden soll, geschlossenen Mietvertrag Hauptmiete mit dem Eigentümer oder den Eigentümern der Liegenschaft zustande, doch geht mit der Begründung von Wohnungseigentum am Mietgegenstand die Rechtsstellung des Vermieters auf den Wohnungseigentümer über.</satz>  
</absatz> [...]
```

Quelle: eigene Darstellung

Abbildung 3.2.: Schritt 2: Durchführung einer manuellen Analyse im RIS-XML (§2 Abs. 1 Satz 1 bis 3 MRG)

welche im Anhang im Detail zu finden ist, ergibt zehn Satzmuster, welche in Tabelle 3.1 enthalten sind.

Nr.	Muster	Regulärer Ausdruck	Anzahl
1	kleiner Buchstabe[.] (incl. ß)	[a-zß][.]	235
2	kleiner Buchstabe[.] Leerzeichen großer Buchstabe (incl. Umlaute)	[a-zß][.] [A-ZÜÖÄ]	214
3	Zahl[.]	[1-9][.]	8
4	kleiner Buchstabe[:](incl. ß)	[a-zß][:]	6
5	Zahl[.] Leerzeichen großer Buchstabe	[1-9][.] [A-ZÜÖÄ]	4
6	kleiner Buchstabe[.] Leerzeichen [§]	[a-zß][.] [§]	4
7	großer Buchstabe[.] Leerzeichen großer Buchstabe	[A-ZÜÖÄ][.] [A-ZÜÖÄ]	3
8	großer Buchstabe[)]][.]	[A-ZÜÖÄ)]][.]	2
9	großer Buchstabe[.]	[A-ZÜÖÄ][.]	2
10)]][.] Leerzeichen großer Buchstabe)]][.] [A-ZÜÖÄ]	1
•	•	•	479

Tabelle 3.1.: Ergebnisse der Satzmusteranalyse

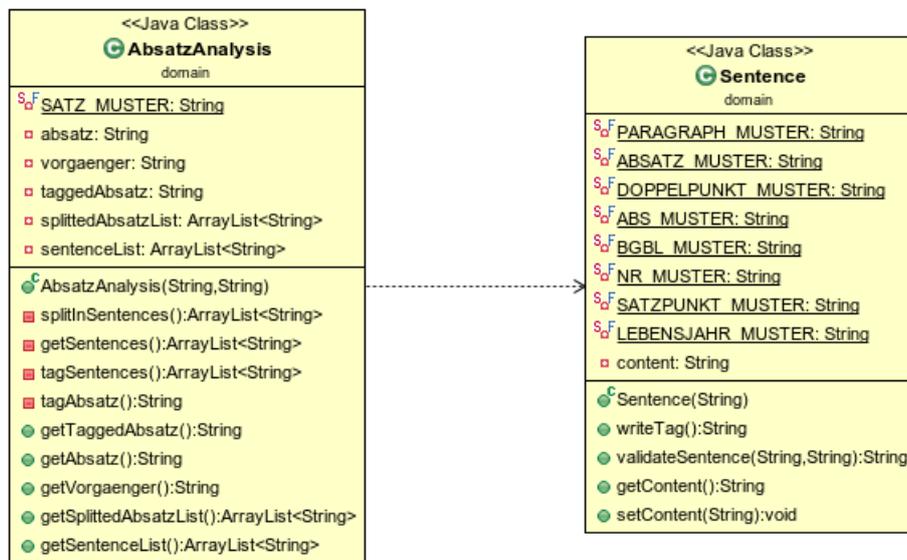
Bei einem Satzmuster handelt es sich um das letzte Zeichen des Satzendes und - falls vorhanden - den ersten zwei Zeichen des darauffolgenden Satzes. Die Satzmuster werden anschließend mit Hilfe von regulären Ausdrücken formal abgebildet. Beispielsweise wird das Satzmuster "kleiner Buchstabe + Punkt" in den regulären Ausdruck "[a-zßäü][.]" übersetzt.

3.3. Automatisierte Analyse

Die erarbeiteten regulären Ausdrücke müssen nun richtig eingesetzt werden, um das Ziel einer automatisierten Satzmustererkennung zu erreichen. Dazu bedarf es zunächst einer Konzeptionsphase, welche sinnbildlich gesprochen einen Rahmen für die Bearbeitung entwirft. Für diese ist der Algorithmus in zwei Klassen aufgeteilt, *AbsatzAnalysis* und *Sentence*, welche in einer 1:n Beziehung zueinander stehen. Abbildung 3.3 beinhaltet die vorgestellten Klassen, ihre Parameter und Methoden in einem Klassendiagramm.

Nun können auf Basis der vorgestellten regulären Ausdrücke eine automatisierte Satzmustererkennung in der Programmiersprache Java entwickelt werden,

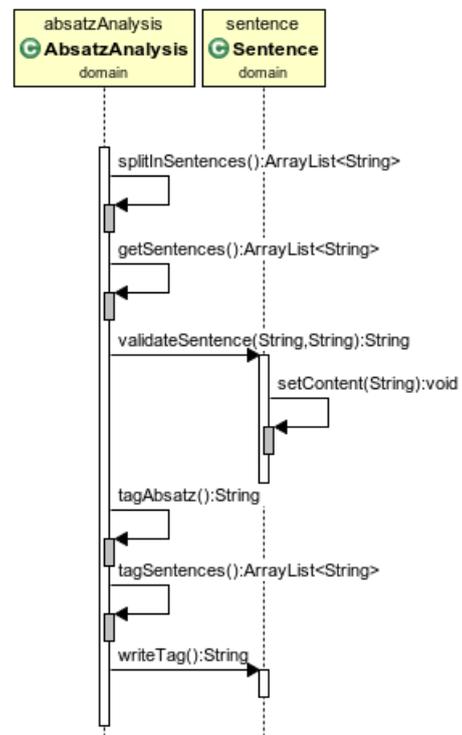
3. Implementierung einer Satzmustererkennung für Gesetzestexte



Quelle: eigene Darstellung

Abbildung 3.3.: Klassendiagramm zur Satzanalyse

welche sich um die automatische Markierung der Sätze im RIS-XML kümmert. Das Zusammenspiel dieser zwei Klassen ist in einem Sequenzdiagramm dargestellt, welches in der Klasse *AbsatzAnalysis* beginnt. Hier wird im ersten Schritt der Inhalt eines Absatzes mit Hilfe der identifizierten Satzmuster in einem regulärem Ausdruck in mögliche Sätze geteilt (*splitInSentence()*) und anschließend in einer *ArrayList splittedAbsatzList* aus Strings gespeichert. Anschließend werden die aufgeteilten Sätze einzeln mit ihrem Vorgänger und Nachfolger analysiert (*getSen-*



Quelle: eigene Darstellung

Abbildung 3.4.: Sequenzdiagramm

tences(), validiert(*validateSentence()*) und erneut in einer *ArrayList sentenceList* aus Strings gespeichert. Abschließend müssen die Sätze bzw. Teilsätze und Fragmente mit dem entsprechendem XML-Tag am Anfang und Ende markiert werden.

Das Ergebnis ist ein RIS-XML mit annotierten Sätzen, wie es bereits in der manuellen Analyse in Abbildung 3.5 zu sehen war. Nach einem Vergleich mit den manuellen Markierungen und einer Evaluierung der Satzmustererkennung, welche in Kapitel 5 nachzulesen ist, wurde die automatisierte Analyse erweitert und Sonderfälle eingeführt, die im nachfolgenden Kapitel dargestellt sind.

3.4. Erweiterungen und Sonderfälle

Zwei weitere Lösungsansätze, die die implementierte Satzmustererkennung optimieren soll, werden entwickelt. Dabei handelt es sich um die Erweiterung der automatischen Satzmustererkennung

- zur Markierung von Teilsätzen
- und einer zusätzlichen Markierung der restlichen Bestandteile des Gesetzes als Fragment, sodass Ausnahmen identifiziert werden können.

```

<absatz typ="abs" ct="text" halign="j">
  <gldsym>§ 10.</gldsym> (1)
  <satz>Der Hauptmieter einer Wohnung, der in den
  letzten zwanzig Jahren vor der Beendigung des
  Mietverhältnisses in der gemieteten Wohnung
  Aufwendungen zur wesentlichen Verbesserung (§ 9)
  gemacht hat, die über seine Mietdauer hinaus
  wirksam und von Nutzen sind, oder der solche
  Aufwendungen dem Vormieter [...]</satz>
  <teilsatz>Das Ausmaß dieser Abschreibung beträgt
  für jedes vollendete Jahr </teilsatz>
</absatz>
<liste>
<ziffernliste ebene="1">
  <listelem ct="text">
    <symbol stellen="2">1.</symbol>
    <teilsatz> bei den in Abs. 3 Z 1 und 3
    genannten Aufwendungen ein Zehntel
    </teilsatz>
  </listelem>
[...]
```

Quelle: eigene Darstellung

Die Markierung von Teilsätzen setzt **Abbildung 3.5.:** Ausschnitt annotierte Teilsätze im RIS-XML (§10 MRG) voraus, dass auch der Vorgänger und Nachfolger eines erkannten Satzes/Satzteiles in Betracht gezogen werden müssen. Ausnahmen, die diese These unterstützen, sind:

- Teilsätze, die mit einem großem Buchstaben beginnen und einem Satzpunkt enden,

- Abkürzungen, die nicht als solche identifiziert werden, beispielsweise „Nr. 1“ oder „II. Hauptstück“
- und Datumsangaben, wie 30. November.

Für Datumsangaben und Abkürzungen konnten des Weiteren zusätzliche Muster bestimmt und eingebaut werden. Ein Fragment wird vom Algorithmus als Teilsatz erkannt, folgt aber kein weiterer Teilsatz, wird dieser als Fragment markiert, sodass durch eine Benutzerinteraktion manuelle Markierungen vorgenommen werden können.

Wie diese Ausnahmen dem Benutzer im Tool angezeigt werden, ist in Kapitel 4.3.1 zu sehen.

3.5. Fazit

Anhand dieses Verfahrens ist zu erkennen, dass es durchaus sinnvoll ist schon in einem früheren Prozessschritt Sätze zu identifizieren bzw. zu markieren, sodass eine nachgelagerte Erkennung von Sätzen sich erübrigt und weiterführende Analysen, auch bezüglich der Semantik, vereinfacht werden. Im RIS-XML könnte dies ein XML-Tag sein, welcher den Satzpunkt erkennen lässt. Folgendes Beispiel zeigt die Notwendigkeit der Einführung von Satzpunktmarkierungen auf: „§45. (1) Im Fall eines vor dem 1. März 1994 geschlossenen Hauptmietvertrags kann der Vermieter den Hauptmietzins für eine Wohnung der Ausstattungskategorie A [...], anheben, [...]. Die angeführten Beträge valorisieren sich entsprechend der Regelung des § 16 Abs. 6. [...]“.

Die Ausschnitte des § 45 Abs. 1 MRG zeigen, dass das Satzzeichen des Punktes nicht eindeutig ein Satzende bestimmt, sondern auch für Datumsangaben, wie hier „1.März 1994“, oder für Abkürzungen, wie beispielsweise „Abs.“ für Absatz, genutzt wird. Ein XML-Tag <satz> im RIS-XML würden diese Unstimmigkeiten aufheben, sodass genauso wie mit der automatisierten Satzmustererkennung Gesetzestexte auf Satzebene zum einen visualisiert, zum anderen referenziert werden können.

4. Modellierungsmethode

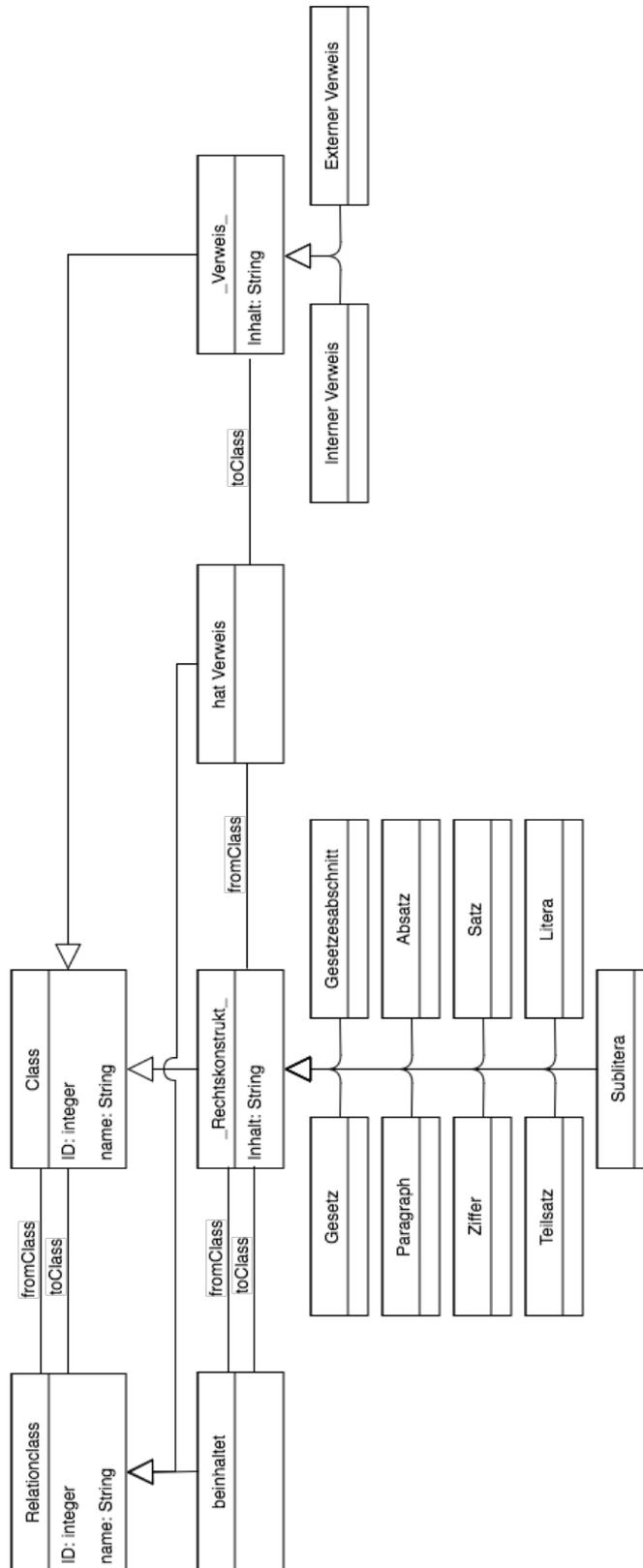
Eines der Ziele der vorliegenden Arbeit ist die Erarbeitung und zur Verfügungsstellung einer Modellierungsmethode zur Visualisierung und Analyse von Gesetzestexten. In diesem Kapitel sind die Details zu:

- dem Metamodell der Methode,
- der Umsetzung der automatischen Generierung von Gesetzesmodellen
- und die Vorstellung der Modelle für eine IT-gestützte Modellierung vom Gesetz nachzulesen.

4.1. Metamodell der Methode

Eine semi-formale Herangehensweise ermöglicht es die Syntax der Modellierungssprache durch ein Metamodell und dessen Semantik in natürlicher Sprache zu definieren. Das Metamodell enthält die in einem Modell möglichen Elemente und ihre Relationen zueinander und ist in Abbildung 4.1 dargestellt. Die Abbildung zeigt das Metamodell der erstellten Modellierungssprache. Die abstrakte Klasse „_Rechtskonstrukt_“ stellt dabei eine Klasse, die Beziehung "beinhaltet" eine Relation dar. Konkrete Modellierungselemente sind: Gesetz, Gesetzesabschnitt, Paragraph, Absatz, Ziffer, Satz, Teilsatz, Litera und Sublitera. Diese können in einer "beinhaltet" Beziehung zueinander stehen.

Zusätzlich ist die Modellierung von Referenzen möglich. Dazu steht die abstrakte Klasse „_Verweis_“ zur Verfügung, von welcher "Interner Verweis" und "Externer Verweis" erbt. Mit dieser Unterscheidung werden Referenzen im



Quelle: eigene Darstellung

Gesetzesmodell klassifiziert. Die Unterscheidung von internen und externen Referenzen ist dadurch bedacht, dass die Referenzen entweder als Hyperlinks dargestellt werden und somit eine *externe* Ressource aufrufen oder als Modell vorliegen und *intern* sind. Rechtskonstrukte können Referenzen enthalten, indem sie mit der "hat Verweis" Beziehung miteinander verknüpft werden.

4.1.1. Notation

In Tabelle 4.1 und 4.2 sind die Notationen der Klassen und die Details dazu zusammengefasst, in Tabelle 4.3 die Notationen der Relationen.

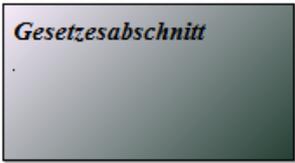
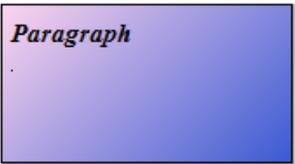
Klasse	Erbt von	Beschreibung	Attribut	Typ
Rechtskonstrukt	Class	Diese abstrakte Klasse fasst alle vorhandenen Rechtskonstrukte zusammen.	Name	String
			Inhalt	Long String
Gesetz 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welches Gesetz es sich handelt.	Name	String
			Inhalt	Long String
Gesetzesabschnitt 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welchen Gesetzesabschnitt es sich handelt.	Name	String
			Inhalt	Long String
Paragraph 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welchen Paragraphen es sich handelt.	Name	String
			Inhalt	Long String

Tabelle 4.1.: Notation der Klassen in ADOxx - Teil 1

4. Modellierungsmethode

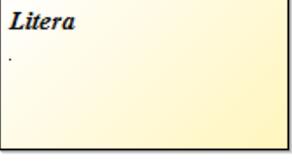
Klasse	Erbt von	Beschreibung	Attribut	Typ
Absatz 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welchen Absatz es sich handelt(falls vorhanden).	Name	String
			Inhalt	Long String
Ziffer 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welche Ziffer es sich handelt (falls vorhanden).	Name	String
			Inhalt	Long String
Satz 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welchen Satz es sich handelt.	Name	String
			Inhalt	Long String
Teilsatz 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welchen Teilsatz es sich handelt(falls vorhanden).	Name	String
			Inhalt	Long String
Litera 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welches Litera es sich handelt(falls vorhanden).	Name	String
			Inhalt	Long String
Sublitera 	_Rechtskonstrukt_	Diese Klasse dient dazu zu beschreiben, um welches Sublitera es sich handelt(falls vorhanden).	Name	String
			Inhalt	Long String

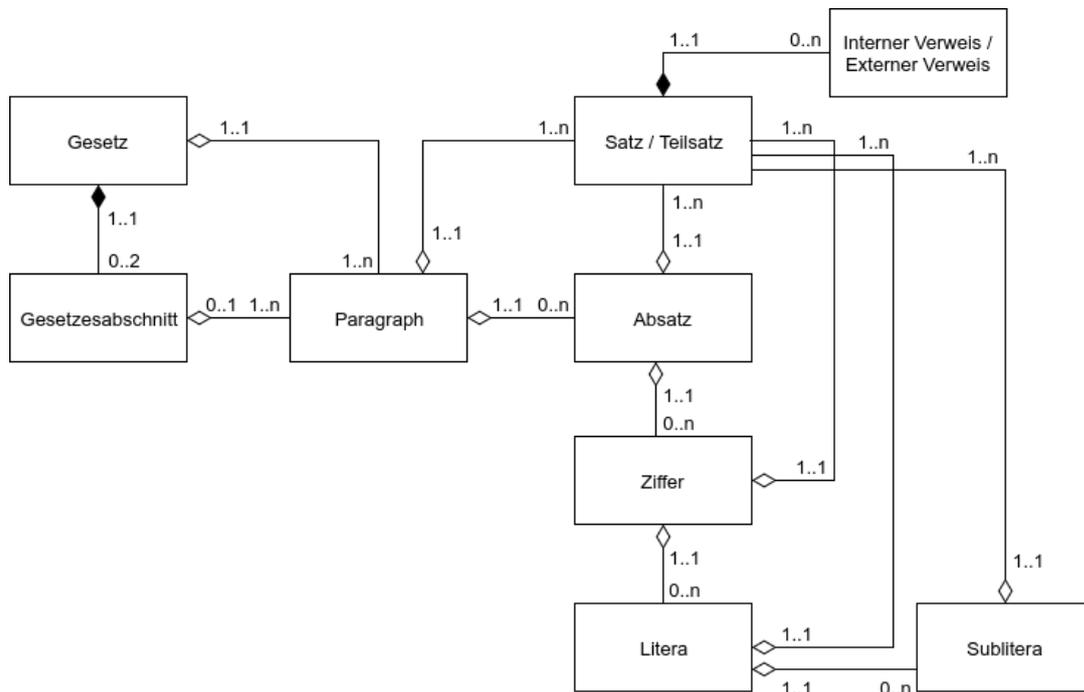
Tabelle 4.2.: Notation der Klassen in ADOxx - Teil 2

Relation	Beschreibung
beinhaltet →	Diese Relation beschreibt die Beziehung zwischen den Rechtskonstrukten.
hat Verweis - - - - ->	Diese Relation beschreibt die Beziehung eines Rechtskonstrukts und eines Verweises.

Tabelle 4.3.: Relationen in ADOxx

4.1.2. Syntax

Die Syntax der Modellierungsmethode ist durch das Metamodell gegeben. Das Klassendiagramm in Abbildung 4.2 zeigt zusätzliche Einschränkungen für die Beziehungen zwischen den Rechtskonstrukten, welche mit Hilfe von Beziehungstypen und Kardinalitäten gesteuert werden können.



Quelle: eigene Darstellung

Abbildung 4.2.: Klassendiagramm mit Kardinalitäten

4.1.3. Semantik

Im ersten Schritt liegt der Fokus für die Modellierungsmethode auf die Abbildung der Struktur der Gesetzestexte. Zukünftige Entwicklungen können sich nun auf die Semantik der abgebildeten Struktur konzentrieren. Eine Möglichkeit wäre hierbei die einzelnen Sätze zu analysieren und mit Hilfe von Natural Language Processing (NLP) die Semantik zu bestimmen und nutzerfreundlich darzustellen. Die Verbindung von Syntax und Semantik der Modelle kann in Anlehnung an Fill (2015) stattfinden.

4.2. Automatische Generierung von Gesetzesmodellen

Zur Visualisierung der Gesetzestexte in Modelle werden die Gesetzestexte in Form von XML-Dateien vom OGD Portal heruntergeladen. Dabei werden die RIS Daten Version 2.2 über die URL http://data.bka.gv.at/ris/v2_2/ogdrisservice.asmx genutzt. Das Format der Dateien für Gesetzestexte wird durchgehend als RIS-XML bezeichnet. Dieses RIS-XML dient als Basis für die automatische Generierung von Gesetzesmodellen. Mit Hilfe des Metamodells können so Modelle automatisch generiert werden.

4.2.1. Automatische Transformation

Für die Darstellung muss das RIS-XML mit Hilfe des Metamodells transformiert werden. Für eine automatische Transformation ist jedoch ein detailliertes Studium des RIS-XML notwendig, um einen korrekten Zugriff auf die XML-Knoten zu gewährleisten und diese entsprechend dem Metamodell einem Gesetzesmodell zu mappen. Details zu diesem Aspekt sind in den folgenden Abschnitten nachzulesen.

RIS-XML

Der Aufbau von Gesetzestexten ist als hierarchisch einzustufen, da sich die Gliederungseinheiten der Gesetze so strukturieren lassen. Jedes Gesetz kann ein oder mehrere Gesetzesabschnitte haben. Jeder Gesetzesabschnitt kann ein oder mehrere weitere Gesetzesabschnitte haben, welche wiederum ein oder mehrere Paragraphen haben muss. So lässt sich diese Reihe mit Absätzen, Ziffern, Litera, Sublitera, Sätzen und Teilsätzen fortführen. Die Syntax des Metamodells, welches den Aufbau von Gesetzestexten widerspiegelt, kann in Abbildung 4.2 nachgesehen werden.



Quelle: Screenshot Altova MapForce

Abbildung 4.3.: RIS-XML Schema

Im Gegensatz zu dieser logisch hierarchischen Struktur ist das RIS-XML listenartig aufgebaut. Die Beschreibung eines RIS-Dokuments bzw. RIS-XML ist in XML Schema Definitionen (XSD) für RIS-Dokumente im OGD Portal festgehalten. In Abbildung 4.4 ist das Schema eines RIS-XML dargestellt. Beim *risdok* handelt es sich um das Wurzelement eines RIS-Dokuments, welches Meta-, Nutz- und Layoutdaten enthält. Die Metadaten geben eine Beschreibung über den jeweiligen Datensatz, bei den Layoutdaten handelt es sich um Doku-

menteinstellungen für Word oder PDF-Dateien (Bundeskanzleramt, 2016b). Die eigentliche Rechtsvorschrift, welche auch den Inhalt für die Transformation zum Metamodell beinhaltet, befindet sich in den Nutzdaten.

Ein beispielhafter Ausschnitt aus dem RIS-XML von §1 MRG Abs. 2 Z 3 ist in Abbildung 4.4 zu sehen. Hier ist der listenartige Aufbau klar erkenntlich. Es beginnt mit einem Knoten <i>absatz</i> , dem ein Knoten <i>liste</i> folgt. Diese <i>liste</i> ist kein Kinds-knoten von <i>absatz</i> , was eine Zugehörigkeit zum Elternknoten einschließen würde. Die Zugehörigkeit zum entsprechenden <i>absatz</i> ist demnach nur durch die Reihenfolge der Knoten herauszufinden. Aus diesem listenartigem Aufbau ist somit nicht klar ersichtlich, um welche Gliederungseinheit es sich im Gesetzestext handelt. Die Problematik stellt eine große Herausforderung dar, da viele Bedingungen erfüllt werden müssen, sodass eine Zuordnung zu einer Gliederungseinheit vom Schema des Gesetzestextes möglich ist.	<pre>1 <absatz typ="abs" ct="text" halign="j"> 2 (2) In den Anwendungsbereich dieses 3 Bundesgesetzes fallen nicht 4 </absatz> 5 <liste> 6 <ziffernliste ebene="1"> 7 [...] 8 <listelem ct="text"> 9 <symbol stellen="2">3.</symbol> 10 Mietverträge, die [...] 11 </listelem> 12 </ziffernliste> 13 <literaliste ebene="2"> 14 <listelem ct="text"> 15 <symbol stellen="2">a</symbol> 16 eine Geschäftsräumlichkeit oder 17 </listelem> 18 <listelem ct="text"> 19 <symbol stellen="2">b</symbol> 20 [...], 21 </listelem> 22 </literaliste> 23 </liste></pre>
---	---

Abbildung 4.4.: RIS-XML: §1 MRG Abs. 2 Z 3

Mapping vom RIS-XML zum Metamodell

Für eine automatische Transformation vom RIS-XML in die Modellierungssprache ist zunächst ein Mapping der relevanten XML-Tags notwendig. Dabei ist von

zwei Schritten die Rede. Im ersten Schritt müssen die für die Transformation relevanten XML-Tags identifiziert werden, da viele Tags im RIS-XML nicht für die Gesetzestexte verwendet werden, sondern auch zur Darstellung der HTML-Seite oder dem Druck eines PDFs dienen.

Element in der Modellierungssprache	Knoten im RIS-XML	Attribut im RIS-XML	Bedingungen im RIS-XML
Gesetzesabschnitt	ueberschrift	ct	equals("text")
		typ	equals("g1")
Gesetzesabschnitt	ueberschrift	ct	equals("text")
		typ	equals("g2")
Paragraph	absatz	ct	equals("artikel_anlage")
Absatz	absatz	ct	equals("text")
Ziffer	Ziffernliste	-	-
Ziffer	Betragliste	-	-
Litera	Literaliste	-	-
Sublitera	Subliteraliste	-	-
Satz	Symbol	-	-
Teilsatz	Symbol	-	-

Tabelle 4.4.: Mapping des RIS-XML zur Modellierungssprache

Im zweiten Schritt müssen diese identifizierten XML-Tags den Elementen in der Modellierungssprache, also dem Metamodell zugeordnet werden. Eine besondere Schwierigkeit stellt die Identifizierung der richtigen Knoten in der korrekten Reihenfolge mit den korrekten Bedingungen dar. Durch den beschriebenen listenartigen Aufbau muss die Abarbeitung der RIS-XML Knoten nacheinander geschehen und dem Metamodell zugeordnet werden. Tabelle 4.4 zeigt einen Überblick über das Mapping des RIS-XML zur Modellierungssprache.

Mit Hilfe eines Java-Programms wurde dann ein Algorithmus entwickelt, welcher die automatische Generierung von Gesetzesmodellen ermöglicht.

4.2.2. Anbindung des RIS

Um den Anforderungen an das System gerecht zu werden und Aspekte wie die Aktualität und Rechtmäßigkeit von Gesetzestexten zu gewährleisten, wurde das RIS über Webservices, welche im OGD Portal zur Verfügung gestellt werden, an die bestehende ADOxx Architektur angebunden. Im Portal stellt der

4. Modellierungsmethode

OgdRisService V2 _2 folgende Methoden bereit:

- "GetDocNumbers
Abrufen von Dokumenten mittels Array von Dokumentnummern
- GetDocNumbersXml
Abfrage mittels Array von Dokumentnummern mit XmlString als Ergebnis
- SearchDocuments
Suche nach Dokumenten mittels OGDrisRequest
- SearchDocumentsXml
Suche nach Dokumenten mittels Xml-String
- Version
Gibt die Versionsnummer des Service zurück" (Bundeskanzleramt, 2016b)

Diese Webservices werden im Editor über AdoScript aufgerufen.

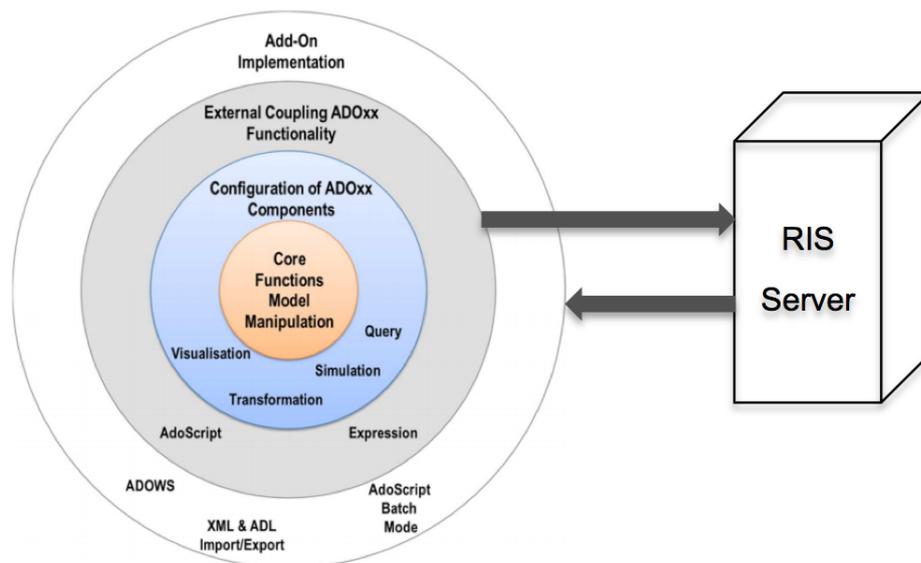


Abbildung 4.5.: ADOxx Architektur mit RIS Anbindung

Der Webservice agiert über SOAP. Dieser beinhaltet eine Suche nach Gesetzestexten und eine Abfrage nach bestimmten Dokumenten, welche in den folgenden Klassen implementiert werden: *SOAPClientDocument* und *SOAPClientRequest*. Über ein SOAPRequest wird wie im OGD Request Schema definiert eine Anfrage nach Bundesnormen gesendet. Als Orientierung dient dabei das Schema

4.2. Automatische Generierung von Gesetzesmodellen

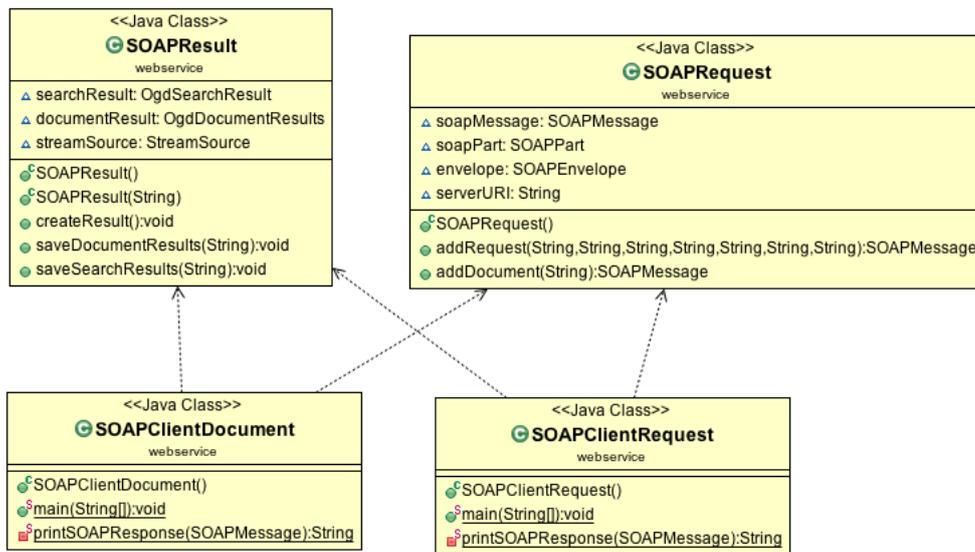


Abbildung 4.6.: Klassendiagramm für SOAP Webservice

BundesnormenSearchRequest. Die SOAP Abfrage fokussiert sich dabei auf die hervorgehobenen Variablen in Abbildung 4.7: Suchworte, Titel, Abschnitt, Fassung, Sortierung, DokumenteProSeite.

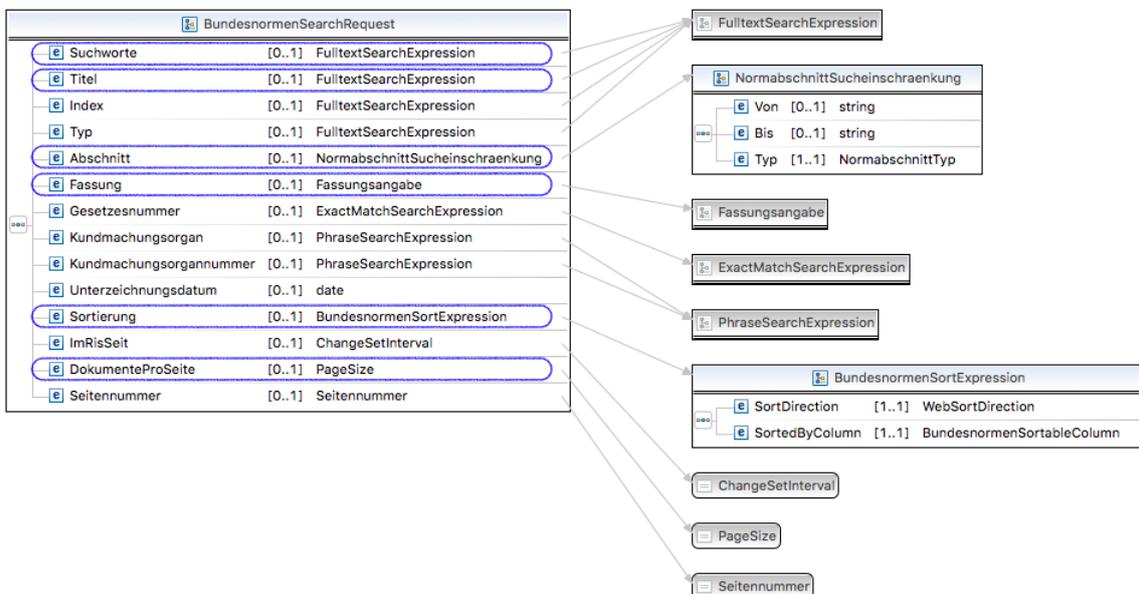


Abbildung 4.7.: Schema zu BundesnormenSearchRequest

In der Klasse *SOAPRequest.java* werden dafür einige Standardeinstellungen vorgenommen, die im Beispielcode von Abbildung 4.8 gezeigt werden.

```
1 PageSize ps = PageSize.ONE_HUNDRED;
2 bundesnormenSearchRequest.setDokumenteProSeite(ps);
3 bundesnormenSearchRequest.setSeitennummer(1);
4 BundesnormenSortExpression bse = new BundesnormenSortExpression();
5 WebSortDirection wsd = WebSortDirection.ASCENDING;
6 bse.setSortDirection(wsd);
7 BundesnormenSortableColumn bsc =
8     BundesnormenSortableColumn.KURZINFORMATION;
9 bse.setSortedByColumn(bsc);
```

Abbildung 4.8.: Definierte Standardeinstellungen für einen SOAPRequest

Bei jeder Suchanfrage über die zur Verfügung gestellte Suche für die Modellierungsmethode wird die erste Seite mit den ersten 100 Suchergebnissen, welche aufsteigend nach den Kurzinformationen sortiert werden, angezeigt.

Für die restlichen Variablen Suchworte, Titel, Abschnitt und Fassung soll der User die Möglichkeit haben die Suchanfrage individuell zu gestalten. Für diesen Zweck hat der User in einem Konfigurationsmodell (4.3.2) Abfragemöglichkeiten.

4.3. Modelle

Für einen Prototypen der Visualisierung wird die Metamodellierungsplattform ADOxx (<http://www.adoxx.org>) herangezogen, welche es ermöglichte den Ansatz der Konzeptualisierung vom Gesetzesmodell darzustellen (Fill und Karagiannis, 2013). Dazu sind verschiedene Funktionalitäten in zwei Modellen umgesetzt:

- dem Gesetzesmodell
- und dem Konfigurationsmodell.

Im Nachfolgenden werden die Modellierungs- als auch Abfragefunktionen in den Modellen detailliert beschrieben.

4.3.1. Das Gesetzesmodell

Das Gesetzesmodell soll es dem Benutzer ermöglichen Gesetzestexte übersichtlich darzustellen. Er hat dabei die Möglichkeit die Anzahl an Gesetzen und/oder Paragraphen pro Modell selbst zu bestimmen. Bei der automatischen Generierung wurde es darauf beschränkt pro Modell einen Paragraphen zu transformieren und visualisieren. Ein beispielhaftes Gesetzesmodell ist in Abbildung 4.9 zu sehen.

Modellierung von Referenzen

Eine der Hauptursachen für das mühsame Verstehen und Analysieren von Gesetzestexten ist die Vielzahl an Referenzen in den Texten. Ein Beispiel aus dem österreichischen Mietrechtsgesetz (MRG) mit hervorgehobenen Verweisungen zeigt die beschriebenen Schwierigkeiten auf:

”§ 6. (1) Unterläßt der Vermieter durchzuführende Erhaltungs- oder Verbesserungsarbeiten, so hat ihm das Gericht (die Gemeinde, § 39) auf Antrag die Vornahme der Arbeiten binnen angemessener, ein Jahr nicht übersteigender Frist aufzutragen. Sind darunter Arbeiten, die nach § 3 Abs. 3 Z 2 lit. a bis c vorweg durchzuführen sind, so ist die Durchführung dieser Arbeiten vorweg aufzutragen; hinsichtlich solcher Arbeiten gilt Abs. 4 nicht. Zur Antragstellung sind berechtigt

- 1. die Gemeinde, in der das Haus gelegen ist, im eigenen Wirkungsbereich und jeder Hauptmieter des Hauses hinsichtlich der im § 3 Abs. 2 Z 1 bis 4 und 6 genannten Erhaltungsarbeiten,*
- 2. die Mehrheit der Hauptmieter – berechnet nach der Anzahl der Mietgegenstände – des Hauses hinsichtlich der im § 3 Abs. 2 Z 5 genannten Erhaltungsarbeiten und der nützlichen Verbesserungen nach Maßgabe des § 4 Abs. 1 und 2[.]”.*

Im Gesetzesmodell hat der User die Möglichkeit Verweisungen auf andere Gesetze zu modellieren. Abbildung 4.10 zeigt ein Beispiel mit einem Ausschnitt

4. Modellierungsmethode

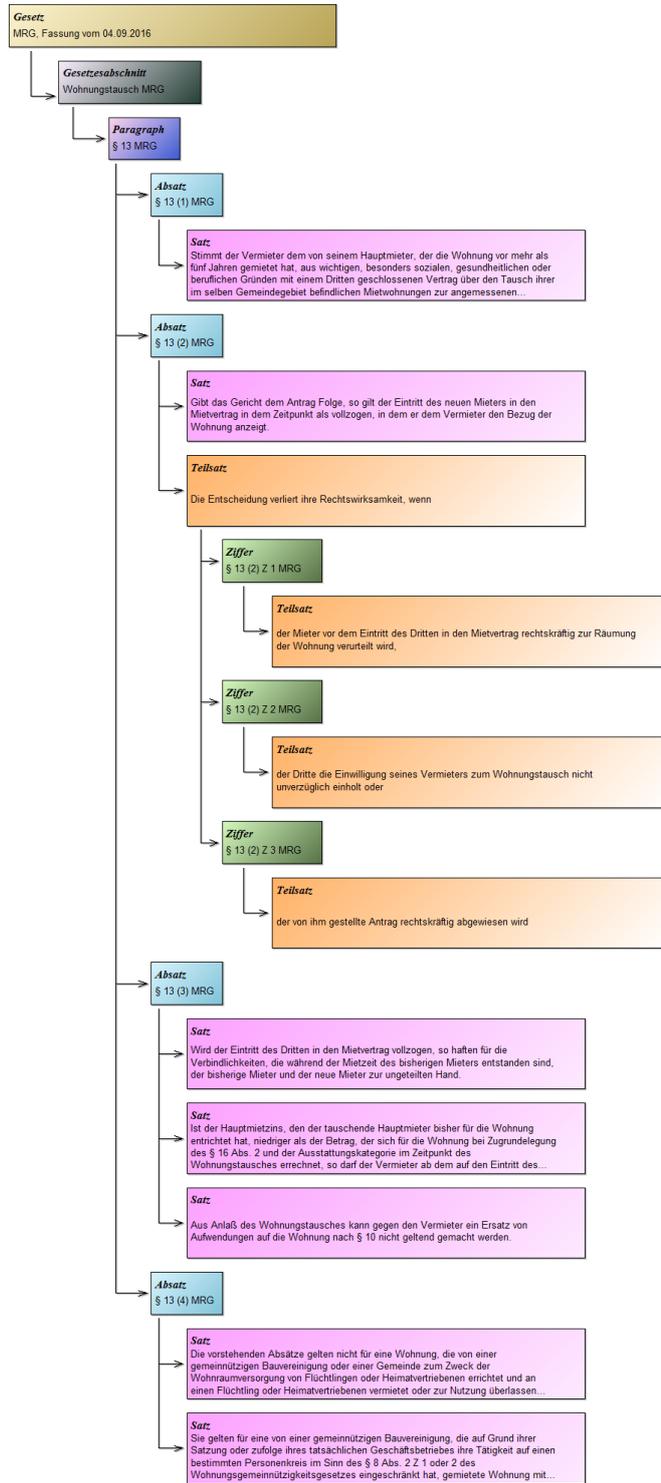


Abbildung 4.9.: Das Gesetzesmodell für § 13 MRG

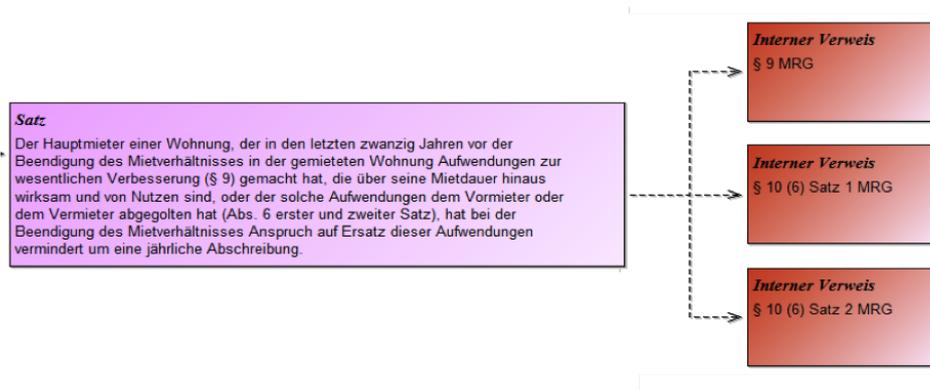


Abbildung 4.10.: Referenzen in § 10 Abs. 1 Satz 1 MRG

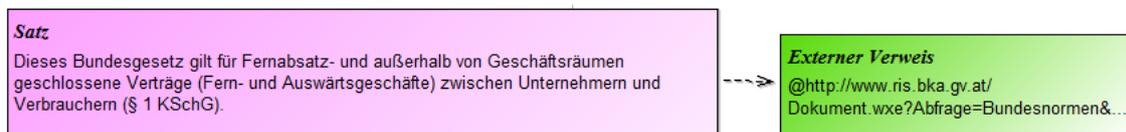


Abbildung 4.11.: Referenz in § 1 Abs. 1 Satz 1 FAGG

aus einem Gesetzesmodell mit internen Verweisen, Abbildung 4.11 mit einem externen Verweis.

Darstellung von Hinweisen

Sätze, die vom entwickelten Programm wie in Kapitel 3 beschrieben, nicht erkannt werden, werden mit Hilfe von Erweiterungen und Sonderfällen (3.4) abgefangen und mit einem Hinweis im Modell dargestellt.

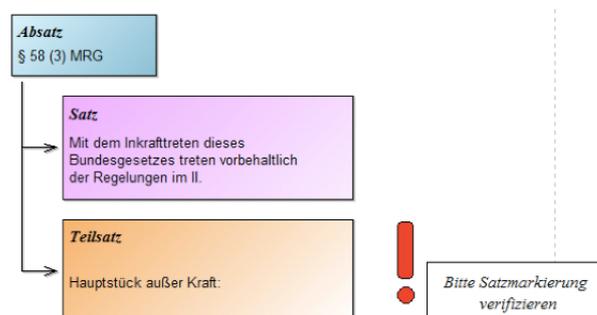


Abbildung 4.12.: Hinweis in § 58 Abs. 3 MRG

Hier wird darauf hingewiesen, dass eine Interaktion des Users erforderlich ist. Dieser hat nämlich die Satzmarkierung zu verifizieren und gegebenenfalls anzupassen, sodass die Aufteilung der Sätze im Gesetzesmodell übereinstimmt. Dieser Hinweis wird nur dargestellt, wenn ein Modell automatisiert generiert wird. Dem User steht somit bei einer manuellen Modellierung dieses Hinweis-Modellelement nicht zur Verfügung.

AQL Abfragen

Durch die Aufteilung der Gesetzestexte bis hin zu Sätzen und Teilsätzen sind nun in der Metamodellierungsplattform ADOxx Analysen möglich. Diese lassen sich mit Hilfe von sogenannten AQL Abfragen realisieren. Dazu müssen lediglich Queries, welche an den individuellen Bedarf anpassbar sind, geschrieben werden.

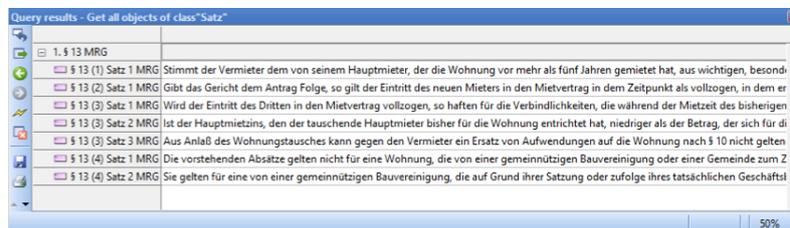


Abbildung 4.13.: Query Ergebnisse

Beispielsweise können auf diese Weise neben der Anzahl der Sätze (siehe Abbildung 4.13) und Teilsätze in einem Paragraphen, auch paragraphenübergreifende Analysen zusammengestellt und ausgewertet werden.

Neben diesen Analysemöglichkeiten können weitergehende Untersuchungen stattfinden, um sowohl Verschachtelungsgrade der Gesetze festzustellen, als auch eine grobe Anzahl an möglichen Verweisen innerhalb und auf andere Gesetze zu schätzen.

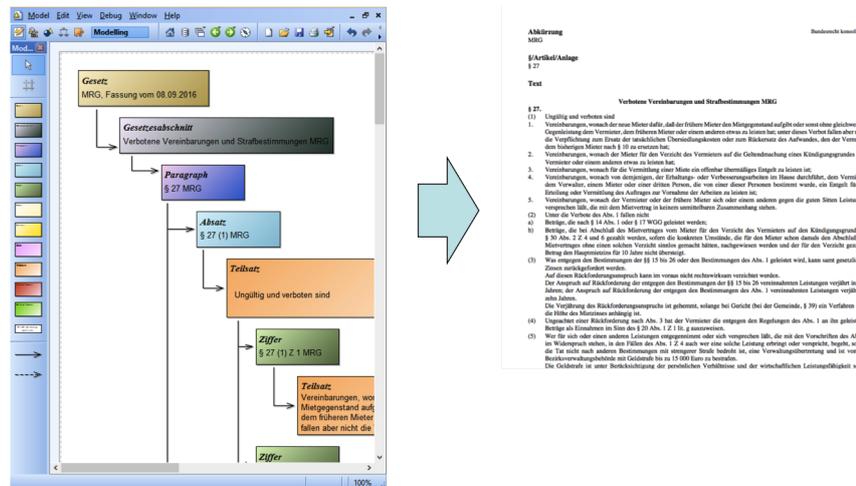


Abbildung 4.14.: PDF-Export

PDF Export

Wie in diesem Kapitel erläutert, wird eine Methode implementiert, um Gesetzestexte als Modelle zu visualisieren. Diese Visualisierung soll, neben der Möglichkeit einer detaillierten Analyse und der Bereitstellung eines Überblick für Laien, auch dazu dienen, Experten gleichermaßen Vorteile zur Verfügung stellen. Diese können sowohl bestehende Gesetze ändern, als auch neue Gesetze entwerfen. Die Verfügbarkeit der Gesetzestexte als Modelle trägt dazu bei, dass Rechtsexperten daran erinnert werden, den Aspekt der Übersichtlichkeit und des Verständnisses für den Laien zu beachten. Als zusätzliches Feature kann demnach eine Möglichkeit geschaffen werden, Gesetzesmodelle aus dem Editor als PDF-Dateien zu exportieren. Somit ergibt sich die Option - analog wie im RIS - Gesetzesentwürfe für den weiteren Gebrauch zu verwenden.

4.3.2. Das Konfigurationsmodell

Um den Einstieg in die Modellierung für Juristen möglichst einfach zu gestalten, dient die Abfrage nach Gesetzen im RIS als Vorbild für ein Modell auf der Modellierungsplattform. Dieses sogenannte Konfigurationsmodell soll ein Wiedererkennungsmerkmal für Rechtsexperten sein und ist an die Suchmaske

Konfigurationsoberfläche
Suchmaske, um Gesetzestexte abzufragen und anschliessend zu visualisieren.

Suchworte

Titel, Abkürzung

Paragraph von bis

Artikel von bis

Fassung vom 

Abbildung 4.15.: Das Konfigurationsmodell

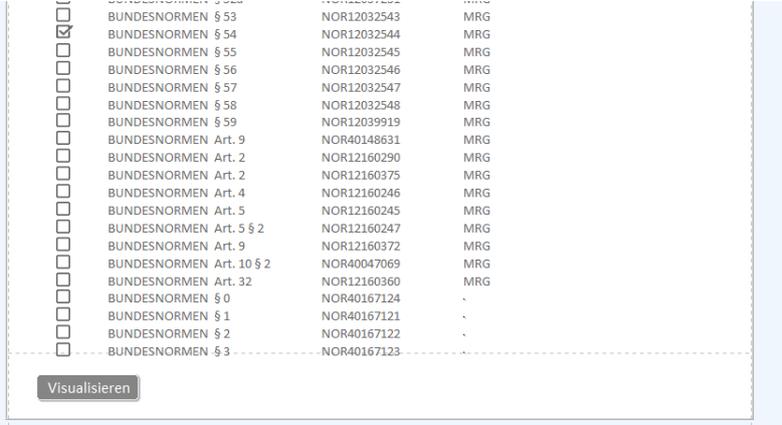
des RIS angelehnt. Eine Suche nach Gesetzestexten ist analog wie im RIS möglich. Abbildung 4.15 zeigt die Umsetzung dessen als Modell. Die Suchfunktionen beschränken sich dabei auf folgende Suchfelder: Suchworte, Titel, Abkürzung, Paragraph von, Paragraph bis, Artikel von, Artikel bis, Fassung vom. Diese sind nach Bedarf und im Laufe einer Weiterentwicklung anpass- und erweiterbar.

Suche nach Gesetzen

Die Suche nach Gesetzen soll sich im Konfigurationsmodell intuitiv gestalten. Nachdem Daten den entsprechenden Feldern der Suchmaske des Konfigurationsmodells eingegeben werden, wird mit einem Klick auf "Suche starten" gesucht. Die Suchergebnisse werden unterhalb der Suchmaske angezeigt. Dabei sind die Ergebnisse auf die ersten 100 Treffer beschränkt und nach den Kurzinformatio- nen sortiert. Zur Bewahrung des Überblicks werden nur Spalten zur Applikation, Paragraphen, Dokumentennummer und Kurzinformatio- n angezeigt (siehe 4.16).

Visualisierung von Suchergebnissen

Für eine Visualisierung eines gesuchten Gesetzestextes wird wie in Abbildung 4.16 auf der linken Seite zu sehen die Checkbox ausgewählt. Eine Schaltfläche



<input type="checkbox"/>	BUNDESNORMEN § 53	NOR12032543	MRG
<input checked="" type="checkbox"/>	BUNDESNORMEN § 54	NOR12032544	MRG
<input type="checkbox"/>	BUNDESNORMEN § 55	NOR12032545	MRG
<input type="checkbox"/>	BUNDESNORMEN § 56	NOR12032546	MRG
<input type="checkbox"/>	BUNDESNORMEN § 57	NOR12032547	MRG
<input type="checkbox"/>	BUNDESNORMEN § 58	NOR12032548	MRG
<input type="checkbox"/>	BUNDESNORMEN § 59	NOR12039919	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 9	NOR40148631	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 2	NOR12160290	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 2	NOR12160375	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 4	NOR12160246	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 5	NOR12160245	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 5 § 2	NOR12160247	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 9	NOR12160372	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 10 § 2	NOR40047069	MRG
<input type="checkbox"/>	BUNDESNORMEN Art. 32	NOR12160360	MRG
<input type="checkbox"/>	BUNDESNORMEN § 0	NOR40167124	-
<input type="checkbox"/>	BUNDESNORMEN § 1	NOR40167121	-
<input type="checkbox"/>	BUNDESNORMEN § 2	NOR40167122	-
<input type="checkbox"/>	BUNDESNORMEN § 3	NOR40167123	-

Visualisieren

Abbildung 4.16.: Visualisierung von Suchergebnissen

"Visualisieren" ermöglicht im Hintergrund

- eine SOAP-Abfrage nach dem ausgewählten Gesetzestext als RIS-XML,
- die automatische Transformation dieses XMLs zum Metamodell
- und den anschließenden Import des Gesetzesmodells im Tool.

5. Evaluierung

In diesem Kapitel wird die Darstellung der Syntax der Gesetzestexte evaluiert. Als Grundlage dient die automatische Satzmustererkennung, welche Ergebnisse sowohl aus der manuellen als auch automatisierten Analyse liefert, die gegenüber gestellt werden können. Dazu wird ein Evaluierungsansatz, eine Basis für eine quantitative Evaluation, Testfälle und eine Precision und Recall Berechnung mit den entsprechenden Ergebnissen vorgestellt.

5.1. Evaluierungsansatz

Die Basis zur Transformation vom RIS-XML zum Metmodell und somit zur Visualisierung von Gesetzestexten schafft die automatisierte Satzmustererkennung, welche als Hauptbestandteil der vorliegenden Arbeit angesehen wird. Für deren Evaluation dienen die Ergebnisse der manuellen und automatisierten Analyse. Dabei wird die manuelle sowie automatische Auszeichnung der Bestandteile des Gesetzes

```
1 <absatz typ="abs" ct="text" halign="j">(2
2   <satz>Untermiete liegt vor, wenn der
3   Mietvertrag mit einer Person geschlossen
4   wird, die in Abs. 1 nicht genannt ist.
5   </satz>
6 </absatz>
```

Abbildung 5.1.: Evaluierungsbeispiel: manuelle Markierung

```
1 <absatz typ="abs" halign="j" ct="text">(2
2 \&lt;satz\&gt;Untermiete liegt vor, wenn der
3 Mietvertrag mit einer Person geschlossen
4 wird, die in Abs. 1 nicht genannt ist.
5 \&lt;/satz\&gt;</absatz>
```

Abbildung 5.2.: Evaluierungsbeispiel: automatisierte Markierung

miteinander verglichen.

Zur Veranschaulichung, wie die Evaluierung stattgefunden hat, ist Abbildung 5.1 und 5.2 ein Beispielparagraph mit den Ergebnissen angebracht. Abbildung 5.1 zeigt § 2 Abs. 2 Satz 1 MRG mit manueller Satzmarkierung, Abbildung 5.2 das Ergebnis der automatisierten Satzmustererkennung. Bei diesem Beispiel liegt der einzige Unterschied darin, dass durch die automatisierte Markierung die Sonderzeichen als Unicode Zeichen dargestellt werden. Diese zwei bearbeiteten XML-Dateien stellen die Grundlage für die Evaluierung dar.

5.1.1. Basis für quantitative Evaluation

Um die Implementierung für die vorliegende Arbeit evaluieren und vergleichen zu können, mussten im ersten Schritt initiale Vergleichswerte definiert werden. Diese haben dazu gedient die Ergebnisse an diesen Werten zu messen und eine Aussage zu fällen, wie gut die gewählte Methode ist. Mit Hilfe dieser Vergleichswerte ergab sich somit ein SOLL-Profil im Gegensatz zu einem IST-Zustand.

Die Tabelle 5.1.1 zeigt die Basiswerte für die quantitative Evaluation. Die Werte beziehen sich nur auf das österreichische Mietrechtsgesetz mit 79 Paragraphen, 479 manuell identifizierten Sätzen und 261 identifizierten Teilsätzen.

Anzahl Paragraphen	79
Anzahl an manuell identifizierten Sätzen	479
Anzahl an manuell identifizierten Teilsätzen	261

Tabelle 5.1.: Basis für quantitative Evaluation des MRG

5.1.2. Testfälle

Die Beschreibung von Testfällen soll es ermöglichen automatisch zu überprüfen, inwieweit das Ergebnis dem SOLL-Profil entspricht. Für diesen Zweck sind Tests

definiert. Für die Tests sind drei Hauptmerkmale der XML-Dateien als Testfälle identifiziert:

- Test auf Gleichheit
- Test nach Unterschieden
- Vergleich des XML Skeletts

Der Test nach Gleichheit überprüft die Strings der XML-Dateien auf Identität. Dabei geht es weniger um die Anzahl an Knoten im XML, sondern eher um die Gleichheit der Zeichen im Dokument.

```
1 @Test
2 public void testAufGleichheit() throws Exception, JAXBException,
3 ParserConfigurationException, SAXException, IOException {
4     assertXMLEqual("Vergleich der beiden XML-Dateien:
5     manuelle und automatisierte Markierung",
6     this.XMLmanuell, this.XMLautomatisch);
7 }
```

Abbildung 5.3.: Unit Testcase Equality

```
1 @Test
2 public void testNachUnterschieden() throws Exception {
3     DetailedDiff detailedDiff = new DetailedDiff(
4     compareXML(this.XMLmanuell, this.XMLautomatisch));
5     @SuppressWarnings("rawtypes")
6     List allDifferences = detailedDiff.getAllDifferences();
7     assertEquals(detailedDiff.toString(), 0,
8     allDifferences.size());
9 }
```

Abbildung 5.4.: Unit Testcase Differences

Der Test nach Unterschieden listet alle Unterschiede auf, die den ersten

```
1 @Test
2 public void vergleichXMLSkelett() throws Exception {
3     DifferenceListener differenceListener =
4     new IgnoreTextAndAttributeValuesDifferenceListener();
5     Diff diff = new Diff(this.XMLmanuell, this.XMLautomatisch);
6     diff.overrideDifferenceListener(differenceListener);
7     assertTrue("Vergleich der beiden XML Skeletts"
8     + diff, diff.similar());
9 }
```

Abbildung 5.5.: Unit Testcase Compare To Skeleton XML

Test nicht bestanden haben, sodass mit Hilfe dieses Tests in der Implementierungsphase weitere Anpassungen vorgenommen werden konnten. Am Schluss steht der Vergleich des XML Skeletts, welcher überprüft, ob das Skelett d.h. die Knoten der XML-Datei identisch sind. Zusätzlich werden hier die Anzahl der Knoten verglichen.

5.1.3. Precision und Recall

Mit den Ergebnissen der vorangehenden Kapitel sollen im abschließenden Schritt Precision und Recall Berechnungen durchgeführt werden. Diese Methode zählt zu einer der am häufigsten genutzten und liefert zudem auch wissenschaftlich vergleichbare Werte.

Precision und Recall, zu deutsch *Genauigkeit* und *Trefferquote* dient der Messung der Informationserschließung und dem Information Retrieval. Die Berechnungen setzen sich aus den vier Bereichen zusammen: true-negative(tn), false-negative(fn), true-positive(tp) und false-positive(fp), welche für den vorliegenden Anwendungsfall wie folgt definiert sind:

tn: Die Markierung der manuellen Analyse sieht keinen Satz vor, ebenso erkennt die automatisierte Analyse keinen Satz.

fn: Die Markierung der manuellen Analyse sieht einen Satz vor, dieser

wird in der automatisierten Analyse jedoch nicht erkannt.

tp: Die Markierung der manuellen Analyse sieht einen Satz vor, dieser wird auch in der automatisierten Analyse erkannt. Die Markierungen der manuellen und automatisierten Analyse stimmen überein.

fp: Die Markierung der manuellen Analyse sieht keinen Satz vor, dieser wird aber in der automatisierten Analyse erkannt.

Recall

Der Recall berechnet sich wie folgt: $\text{Recall} = \frac{tp}{tp+fn}$

Dabei ist tp die Anzahl der gefundenen, relevanten Dokumente, und $tp+fn$ die Anzahl aller relevanten Dokumente in der Datenbank.

Precision

Precision ist folgendermaßen definiert: $\text{Precision} = \frac{tp}{tp+fp}$

Auch hier steht tp für die Anzahl der gefundenen, relevanten Dokumente, $tp+fp$ für die Anzahl aller gefundenen Dokumente. In der vorliegenden Arbeit werden Sätze bzw. Teilsätze im Gesetz als Dokument für die Berechnung definiert.

5.2. Ergebnisse

Die Ergebnisse der Evaluierungsmöglichkeiten werden in den nachfolgenden Kapitel präsentiert.

5.2.1. Testergebnisse

Eine detaillierte Ergebnistabelle über die Tests ist im Anhang zu finden. Die Interpretation der Ergebnisse findet in den nächsten zwei Bereichen 5.2.2 und 5.2.3 statt.

5.2.2. Quantitative Evaluation

Die Tabelle 5.2 zeigt eine quantitative Evaluation mit den in 5.1.1 definierten Basiswerten.

Vergleichsparameter	manuell	automatisiert	Prozentsatz
Anzahl Paragraphen mit erfolgreichem Test	79	69	87,34%
Anzahl an Sätzen	479	469	97,91%
Anzahl an Teilsätzen	261	239	91,57%

Tabelle 5.2.: Quantitative Evaluation des MRG

Die korrekte Abdeckung und Erkennung der automatischen Satzmustererkennung liegt dabei bei 87% der Paragraphen und somit der richtig dargestellten Gesetzesmodelle. Der restliche Teil beinhaltet fehlerhafte Satzmarkierungen wie

- Teilsätze, die mit einem großem Buchstaben beginnen und einem Satzpunkt enden,
- Abkürzungen, die nicht als solche identifiziert werden, beispielsweise "Nr. 1" oder "II. Hauptstück",
- und Datumsangaben, wie 30. November.

Für den restlichen Teil der 13% wurden im zweiten Schritt zwei weitere Lösungsansätze entwickelt, die bereits in Kapitel 3.4 genauer erläutert werden.

5.2.3. Precision und Recall Ergebnisse

Zur Berechnung von Precision und Recall werden folgende Werte herangezogen, die sich aus der quantitativen Evaluation ergeben. Hier soll darauf hingewiesen werden, dass die Werte sowohl von den Sätzen als auch von den Teilsätze zusammengefasst werden. Insgesamt gibt es somit 740 Sätze und Teilsätze zu identifizieren.

Variable	Wert
tn	-
fn	32
tp	708
fp	6

Tabelle 5.3.: Werte zur Berechnung von Precision und Recall

Das Ergebnis für Recall und Precision lässt sich demnach folgendermaßen berechnen:

$$\text{Recall} = \frac{tp}{tp+fn} = \frac{708}{708+32} = 0,9567$$

$$\text{Precision} = \frac{tp}{tp+fp} = \frac{708}{708+6} = 0,9915$$

Der Idealfall für Recall und Precision liegt bei 1. Demnach sind die Ergebnisse für die automatische Satzmustererkennung als sehr gut einzustufen. Besonders beim Ergebnis für Precision entspricht der Wert 0,9915 fast dem Idealwert. Das bedeutet, dass 99,15% relevante Ergebnisse im Suchergebnis vorliegen und mit einer Wahrscheinlichkeit von 0,85% ein fehlerhafter Satz markiert wird. Im Vergleich dazu misst Recall die Wahrscheinlichkeit der relevanten Ergebnisse im Verhältnis zu den möglichen Treffern. Dieser Wert ist mit 95,67% ebenfalls hoch.

6. Conclusio

In diesem Kapitel befindet sich eine Zusammenfassung der vorliegenden Arbeit. Zusätzlich enthält es einen Ausblick für weiterführende Arbeiten und Verbesserungen.

6.1. Zusammenfassung

Im Kapitel 2, der Grundlagen und verwandten Arbeiten, werden einige grundlegende Definitionen und Forschungsbereiche, vor allem auch vorhandene Visualisierungsansätze aufgezeigt. Darauf aufbauend wird in Kapitel 3 eine automatisierte Satzmustererkennung für Gesetzestexte implementiert und mit Hilfe dieser eine Modellierungsmethode in Kapitel 4 entwickelt. Die Modellierungsmethode dient zur Visualisierung und Analyse von Gesetzestexten und soll es verschiedenen Nutzergruppen ermöglichen die Arbeit mit Gesetzen zu optimieren. Rechtsexperten und Legisten haben die Möglichkeit sich einen Überblick über ihre Arbeit mit Gesetzestexten zu schaffen und Defizite bezüglich der Verständlichkeit und dem Überblick der verfassten Texten zu minimieren. Rechtsexperten haben die zusätzliche Möglichkeit während ihrer Arbeit mit Mandanten die Modellierungsmethode zu verwenden. Rechtlichen Laien wird es ermöglicht sich mit Gesetzestexten auseinander zu setzen und vor allem zu verstehen. Im Unternehmen können Gesetzesmodelle in der Geschäftsprozessmodellierung in Hinsicht auf beispielsweise Compliance Regelungen unterstützend eingesetzt werden und zu Prozessoptimierungen führen.

Zum Abschluss ist in Kapitel 5 eine Evaluierung der Satzmustererkennung zu finden, welche mit deren Ergebnissen den umgesetzten Ansatz unterstützt und

das Ziel der Arbeit bestätigt.

6.2. **Ausblick**

Spätere Arbeiten können einen Vergleich verfügbarer Tools und Methoden für eine Satzmustererkennung enthalten. Dabei sind besonders das Verhalten und die Ergebnisse interessant, welche der, in dieser Arbeit durchgeführten, manuellen Analyse gegenübergestellt, analysiert und verglichen werden können.

Zusätzlich ist eine Erweiterung der automatischen Transformation in vielen verschiedenen Formen möglich. Zum einen können und sollten Tests und Anpassungen auf andere Gesetzestexte als das MRG, sowie auf andere Applikationen als das Bundesrecht konsolidiert stattfinden, zum anderen könnte der semantische Aspekt der Modellierungsmethode erweitert werden. Auch die meist kompliziert gestalteten Verweisungen auf andere Gesetze oder Gesetzesteile würden zu einer Reduzierung der Komplexität und einem einfacheren Umgang mit dem Gesetz führen. Ein erster Ansatz zur Erkennung und Identifizierung ist in den Implementierung dieser Arbeit teilweise zu finden. Da jedoch der Rahmen einer Masterarbeit bei der Bearbeitung dieses zusätzlichen Aspekts gesprengt werden würde, wurde der Fokus auf die vorgestellten und gleichzeitig höher priorisierten Aspekte gelegt und somit Möglichkeiten für weitere Arbeiten eröffnet. Die Satzanalyse und die Darstellung der Gesetze auf Satzebene kann nun als Basis für Modellerweiterungen mit Verweisungen genutzt werden, welche ebenfalls auf Satzebene vorhanden sind.

Und nicht zuletzt gibt es auch Potenzial bei der Verbesserung der Ansätze und Verfahren dieser Arbeit. Beispiel hierfür wäre die komfortablere Gestaltung der Korrektur der nicht korrekt erkannten Teil-/Sätze oder Erweiterungen der Funktionalitäten bei der Bearbeitung der Gesetzesmodelle. In diesem Zusammenhang könnte eine Studie über die Benutzerfreundlichkeit durchgeführt werden, welche die Modellierungsmethode für die Endnutzer verbessert und zudem verschiedene Aspekte aufdeckt, welche in dieser Arbeit nicht beachtet wurden.

Zusammenfassend lässt sich sagen, dass diese Arbeit einen Beitrag dazu leistet, Potentiale der Rechtsvisualisierung aufzuzeigen, und den ersten Meilenstein für weitere, zukünftige Entwicklungen legt.

A. Abstract

The importance of legal requirements in companies has grown enormously, not at last in the area of information technology. Legislative violations, i.e. not fulfilling the obligations to forward important information, impose penalties from warning expenses to damage claims (Knackstedt, Brelage und Kaufmann, 2006). In order to ensure compliance with the relevant standards in business processes, an exact knowledge of the situation and the laws is required. In particular, there must be an understanding of the law and the corresponding legal texts, which is a great challenge for an ordinary person.

Laws are strictly structured and there are precise guidelines that must be observed by legal experts when working with laws (Bundeskanzleramt, 1990). The resulting structure of the laws reveals a whole series of possible levels of abstraction and structure, beginning with the section and paragraphs, up to digits and litera. Inevitably, the question arises whether legal texts in their textual form provide an adequate overview and therefore contribute optimally to the understanding of the content. Although, according to Richtlinie 9 of Bundeskanzleramt (1990, p. 9) legislation has to be easy to read, the study of a legal rule is made more difficult by a large number of references to other laws, paragraphs, and digits, which lead to poorer readability and an additional increase in complexity.

Especially in the field of information systems, the modeling of compliance regulations is an important area (Rinderle-Ma, Ly und Dadam, 2008). Schleicher et al. (2010) emphasize that in business process modeling, the focus is on a correct modeling of the business process and not in compliance with compliance regulations. For this reason, there should be appropriate ways to reduce com-

plexity by using IT techniques, make them accessible to the user and assist them in the creation of business processes.

This problem is addressed in this master thesis and a solution is presented by developing a modeling method for the visualization of legal texts. Models serve as the abstraction and generalization of the real world (Stachowiak, 1973) and can help both humans and machines to understand and process a subject area. Because of these properties, they play an important role in computer science in a wide range of applications. The goal of the modeling method developed in this thesis is to provide input to legal experts and to assist them in their work with legal texts. The essential concept of this thesis is the possibility of recognizing quality characteristics from generated data and metrics as well as passing these quality features as practical improvement proposals to legal experts. Due to the visual aspect, the method developed can also significantly facilitate the modification of existing laws. The Austrian legal information system (Rechtsinformationssystem (RIS)), which was established in 1980 and has been online since 1997, provides documentation on Austrian law and a free source of information for laws. With the presented solution a better overview can be realized, whereby this work is based on current relevance and is also of public interest.

B. Satzmustererkennung für Gesetzestexte

B.1. Manuelle Satzmusteranalyse

Die folgende Tabelle zeigt die manuelle Analyse der Satzmuster zum MRG. Das Ergebnis der Analyse waren verschiedene Regular Expression für die automatische Satzmustererkennung.

Nr.	Paragraph	Absatz	Ziffer	vor dem Satz	Satzende (1)	Satzanfang (2)	nachfolgender Satz	Muster (1) + (2)	RegEx Gruppe
1	1	1		-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
2	1	3		-3	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
3	1	5		-5	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
4	2	1		-1	d.	S	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
5	2	1		d.	t.	W	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
6	2	1		t.	r.	A	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
7	2	1		r.	t.	E	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
8	2	1		t.	e.	S	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
9	2	1		e.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
10	2	2		-2	t.	W	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
11	2	2		t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
12	2	3		-3	n.	L	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
13	2	3		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
14	3	1		-1	n.	I	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2
15	3	1		n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
16	3	3		-3	n.	R	ja	kleiner Buchstabe[.] Leerzeichen größer Buchstabe	Gruppe 2

B. Satzmustererkennung für Gesetzestexte

17	3	3	n.	s:	leer	nein	kleiner Buchstabe[.]	Gruppe 4
18	3	3	1.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 4
19	4	1	-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
20	4	4	-4	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
21	4	5	-5	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
22	5	1	-1	A.	leer	nein	großer Buchstabe[.]	Gruppe 11
23	5	2	-2	t.	Z	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
24	5	2	t.	s.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
25	5	2	s.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
26	5	3	-3	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
27	6	1	-1	n.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
28	6	1	n.	t.	Z	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
29	6	1a	(1a)	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
30	6	2	-2	n.	Z	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
31	6	2	n.	r.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
32	6	2	r.	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
33	6	2	n.	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
34	6	2	n.	n.	Ü	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe(Umlaut!)	Gruppe 5
35	6	2	n.	t.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
36	6	2	t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
37	6	4	-4	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
38	6	4	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
39	7	1	-1	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
40	7	1	n.	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
41	7	2	-2	6.	Z	ja	Zahl[.] Leerzeichen großer Buchstabe	Gruppe 6
42	7	2	6.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
43	8	1	-1	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
44	8	1	n.	t.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
45	8	1	t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
46	8	3	-3	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
47	9	3	-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
48	10	1	-1	g.	D	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B.1. Manuelle Satzmusteranalyse

49	10	2		n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
50	10	4a	-2 (4a)	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
51	10	4a	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
52	10	6	-6	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
53	10	6	n.	t.	V	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
54	10	6	t.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
55	10	7	-7	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
56	10	8	-8	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
57	11	1	-1	t.	E	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
58	11	2	-2	g.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
59	12	1	-1	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
60	12	1	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
61	12	1	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
62	12	2	-2	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
63	12	2	n.	n.	M	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
64	12	2	n.	g.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
65	12	3	-3	u.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
66	12	3	u.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
67	12a	1	-1	n.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
68	12a	1	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
69	12a	1	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
70	12a	2	-2	n.	Ä	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe(Umlaut!)	Gruppe 5
71	12a	2	n.	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
72	12a	2	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
73	12a	3	-3	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
74	12a	3	t.	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
75	12a	3	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
76	12a	4	-4	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
77	12a	5	-5	n.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
78	12a	5	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B. Satzmustererkennung für Gesetzestexte

79	12a	5		n.	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
80	12a	5		n.	n.	Ä	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe(Umlaut!)	Gruppe 5
81	12a	5		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
82	12a	6		-6	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
83	12a	7		-7	n.	Z	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
84	12a	7		n.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
85	12a	8		-8	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
86	12a	8		n.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
87	13	1		-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
88	13	2		-2	t.	D	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
89	13	3		-3	d.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
90	13	3		d.	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
91	13	3		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
92	13	4		-4	t.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
93	13	4		t.	ß.	leer	nein	kleiner Buchstabe[.] Buchstabe(ß)[.]	Gruppe 8
94	14	1		-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
95	14	2		-2	n.	M	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
96	14	2		n.	d.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
97	14	2		d.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
98	14	3		-3	n.	L	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
99	14	3		n.	t.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
100	14	3		t.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
101	15	2		-2	t.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
102	15	2		t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
103	15	3		-3	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
104	15	3		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
105	15	4		-4	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
106	15	4		t.	e.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
107	15	4		e.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B.1. Manuelle Satzmusteranalyse

108	15	4	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
109	15a	2	-2	s.	E	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
110	15a	2	s.	d.	I	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
111	15a	2	d.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
112	15a	4	-4	n.	D	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
113	15a	4	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
114	16	2	-2	t.	F	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
115	16	2	t.	d:	leer	nein	kleiner Buchstabe[:]	Gruppe 4
116	16	3	-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
117	16	4	-4	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
118	16	5	-5	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
119	16	6	-6	n.	B	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
120	16	6	n.	n.	D	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
121	16	6	n.	n.	D	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
122	16	6	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
123	16	7	-7	H.	W	ja	großer Buchstabe[.]	Gruppe 9
							Leerzeichen großer Buchstabe	
124	16	7	H.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
125	16	8	-8	t.	D	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
126	16	8	t.	n.	B	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
127	16	8	n.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
128	16	9	-9	m.	B	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
129	16	9	m.	t.	E	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
130	16	9	t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
131	16	10	-10	n.	S	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
132	16	10	n.	n.	B	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
133	16	10	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
134	16	11	-11	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
135	16	12	-12	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
136	16a	1	-1	m.	D	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	
137	16a	1	m.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
138	16a	2	-2	r.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
139	16b	1	-1	n.	W	ja	kleiner Buchstabe[.]	Gruppe 2
							Leerzeichen großer Buchstabe	

B. Satzmustererkennung für Gesetzestexte

140	16b	1	n.	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
141	16b	1	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
142	16b	2	-2	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
143	16b	3	-3	n.	N	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
144	16b	3	n.	IO).	leer	nein	großer Buch- stabe)]].]	Gruppe 10
145	16b	4	-4	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
146	17	1	-1	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
147	17	1a	(1a)	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
148	17	1a	n.	n.	K	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
149	17	1a	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
150	17	1a	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
151	17	1a	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
152	17	2	-2	n).	K	ja)]].] Leerzeichen großer Buchstabe	Gruppe 13
153	17	2	n).	n.	V	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
154	17	2	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
155	17	3	-3	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
156	17	3	n.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
157	18	1	-1	n.	Z	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
158	18	2	-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
159	18	3	-3	f.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
160	18	4	-4	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
161	18a	1	-1	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
162	18a	2	-2	t.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
163	18a	2	t.	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
164	18a	2	n.	n.	H	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
165	18a	2	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
166	18b		§ 18b.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
167	18b		n.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
168	18c	1	-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
169	18c	2	-2	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
170	18c	3	-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
171	18c	4	-4	d.	R	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
172	18c	4	d.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1

B.1. Manuelle Satzmusteranalyse

173	19	1	-1	g.	Z	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
174	19	1	g.	t.	D	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
175	19	2	-2	n.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
176	19	2	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
177	19	3	-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
178	20	1	-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
179	20	2	-2	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
180	20	3	-3	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
181	20	3	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
182	20	4	-4	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
183	20	4	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
184	20	4	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
185	20	5	-5	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
186	21	2	-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
187	21	3	-3	f.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
188	21	3	f.	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
189	21	3	n.	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
190	21	3	n.	d.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
191	21	3	d.	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
192	21	3	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
193	21	4	-4	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
194	21	4	n.	d.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
195	21	4	d.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
196	21	5	-5	4.	leer	nein	Zahl[.]	Gruppe 3
197	21	6	-6	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
198	22		§ 22.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
199	23	1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
200	24	1	-1	17.	leer	nein	Zahl[.]	Gruppe 3
201	24	2	-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
202	24	2a	(2a)	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
203	24	2a	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
204	24	3	-3	ß.	leer	nein	kleiner Buchstabe[.]	Gruppe 8
205	25		§ 25.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1

B. Satzmustererkennung für Gesetzestexte

206	26	1	-1	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
207	26	1	n.	d.	§	ja	kleiner Buchstabe[.] Leerzeichen [§]	Gruppe 7
208	26	1	d.	ß.	leer	nein	kleiner Buchstabe(ß)[.]	Gruppe 8
209	26	2	-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
210	26	3	-3	H.	W	ja	großer Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 9
211	26	3	H.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
212	26	4	-4	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
213	26	4	t.	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
214	26	4	n.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
215	27	3	-3	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
216	27	3	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
217	27	3	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
218	27	3	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
219	27	4	-4	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
220	27	5	-5	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
221	27	5	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
222	27	5	n.	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
223	27	5	n.	e.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
224	27	5	e.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
225	27	6	-6	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
226	27	7	-7	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
227	28		§ 28.	t.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
228	28		t.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
229	28		n.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
230	29	2	-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
231	29	3	(3) a)	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
232	29	3	b)	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
233	29	3	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
234	29	4	-4	n.	N	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
235	29	4	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
236	30	1	-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1

B.1. Manuelle Satzmusteranalyse

237	30	3	-3	m.	Ü	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe(Umlaut!)	Gruppe 5
238	30	3	m.	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
239	30	3	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
240	31	1	-1	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
241	31	1	t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
242	31	2	-2	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
243	31	2	n.	g.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
244	31	3	-3	l.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
245	31	3	l.	t.	M	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
246	31	3	t.	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
247	31	3	t.	n.	leer	nein	kleiner Buchstabe[.]	Punkt fehlt —>hinzuge- fügt- Gruppe 1
248	31	4	-4	t.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
249	31	4	t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
250	31	5	-5	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
251	31	6	-6	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
252	31	6	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
253	32	1	-1	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
254	32	1	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
255	32	2	-2	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
256	32	2	n.	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
257	32	2	t.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
258	32	2	n.	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
259	32	2	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
260	32	3	-3	n.	K	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
261	32	3	n.	n.	D	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
262	32	4	-4	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1

B. Satzmustererkennung für Gesetzestexte

263	32	5	-5	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
264	32	5	n.	t.	Z	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
265	32	5	t.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
266	32	5	n.	12.	leer	nein	Zahl[.]	Gruppe 3
267	33	1	-1	n.	G	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
268	33	1	n.	ZPO.	D	ja	großer Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 9
269	33	1	ZPO.	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
270	33	1	n.	t.	G	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
271	33	1	t.	g.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
272	33	2	-2	e.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
273	33	2	e.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
274	33	3	-3	ABGB).	leer	nein	großer Buch- stabe)]].]	Gruppe 10
275	33a		§ 33a.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
276	33a		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
277	34	1	-1	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
278	34	1	t.	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
279	34	1	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
280	34	2	-2	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
281	34	3	-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
282	34a	1	-1	l.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
283	34a	1	l.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
284	34a	2	-2	l.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
285	34a	2	l.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
286	34a	3	-3	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
287	35	1	-1	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
288	35	1	n.	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
289	35	1	n.	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
290	35	1	n.	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B.1. Manuelle Satzmusteranalyse

291	35	1		n.	2.	leer	nein	Zahl[.]	Gruppe 3
292	35	2		-2	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
293	35	3		-3	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
294	36			§ 36.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
295	37	1		-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 4
296	37	2		-2	g.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
297	37	2a		(2a)	u.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
298	37	3		-3	n:	leer	nein	kleiner Buchstabe[.]	Gruppe 4
299	37	3	1	1.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
300	37	3	2	2.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
301	37	3	3	3.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
302	37	3	4	4.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
303	37	3	4	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
304	37	3	4	n.	g.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
305	37	3	4	g.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
306	37	3	5	5.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
307	37	3	6	6.	i.	Ü	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe(Umlaut!)	Gruppe 5
308	37	3	6	i.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
309	37	3	7	7.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
310	37	3	8	8.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
311	37	3	9	9.	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
312	37	3	9	n.	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
313	37	3	9	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
314	37	3	10	10.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
315	37	3	11	11.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
316	37	3	12	12.	s.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
317	37	3	12	s.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
318	37	3	13	13.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
319	37	3	14	14.	n.	§	ja	kleiner Buchstabe[.] Leerzeichen [§]	Gruppe 7
320	37	3	14	n.	n.	§	ja	kleiner Buchstabe[.] Leerzeichen [§]	Gruppe 7
321	37	3	14	n.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
322	37	3	15	15.	n.	F	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
323	37	3	15	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
324	37	3	16	16.	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
325	37	3	16	t.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
326	37	3	16	n.	n.	Z	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
327	37	3	16	n.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
328	37	3	17	17.	e.	H	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B. Satzmustererkennung für Gesetzestexte

329	37	3	17	e.	z.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
330	37	3	17	z.	H.	leer	nein	großer Buchstabe[.]	Gruppe 11
331	37	3	18	18.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
332	37	3	19	19.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
333	37	3	20	20.	n.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
334	37	3	20	n.	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
335	37	3	20	n.	ß.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 12
336	37	3	20	ß.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
337	37	4	-4	-4	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
338	39	1	-1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
339	39	2	-2	-2	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
340	39	3	-3	-3	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
341	39	3	n.	n.	1991.	leer	nein	Zahl[.]	Gruppe 3
342	39	4	-4	-4	n.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
343	39	4	n.	n.	g.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
344	39	5	-5	-5	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
345	40	1	-1	-1	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
346	40	1	n.	n.	t.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
347	40	1	t.	t.	d.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
348	40	1	d.	d.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
349	40	2	-2	-2	t.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
350	40	2	t.	t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
351	40	3	-3	-3	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
352	40	3	n.	n.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
353	40	3	n.	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
354	41		§ 41.	§ 41.	e.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
355	41		e.	e.	l.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
356	42	1	-1	-1	n.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
357	42	1	n.	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
358	42	2	-2	-2	g.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
359	42	2	g.	g.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B.1. Manuelle Satzmusteranalyse

360	42	2		n.	m.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
361	42	2		m.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
362	42	3		-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
363	42	4		-4	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
364	42	5		-5	r.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
365	42	6		-6	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
366	42a	1		-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
367	42a	2		-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
368	42a	3		-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
369	43	1		-1	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
370	43	2		-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
371	45	1		-1	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
372	45	1		t.	6.	D	ja	Zahl[.] Leerzeichen großer Buchstabe	Gruppe 6
373	45	1		6.	t.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
374	45	1		t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
375	45	2		-2	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
376	45	2		n.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
377	45	3		-3	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
378	45	3		n.	z.	I	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
379	45	3		z.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
380	46	1		-1	e.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
381	46	1		e.	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
382	46	2		-2	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
383	46	2		t.	6.	I	ja	Zahl[.] Leerzeichen großer Buchstabe	Gruppe 6
384	46	2		6.	d.	G	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
385	46	2		d.	d.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
386	46	2		d.	t.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
387	46	2		t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
388	46a	1		-1	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
389	46a	2		-2	t.	S	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
390	46a	2		t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
391	46a	3		-3	e.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
392	46a	3		e.	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B. Satzmustererkennung für Gesetzestexte

393	46a	3		n.	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
394	46a	5		-5	t.	M	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
395	46a	5		t.	r.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
396	46a	6		-6	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
397	46a	6		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
398	46b			§ 46b.	r.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
399	46b			r.	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
400	46b			n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
401	46c			§ 46c.	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
402	46c			t.	t.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
403	46c			t.	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
404	47	1		-1	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
405	47	2		-2	n.	E	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
406	48	1		-1	n.	W	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
407	48	1		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
408	48	2		-2	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
409	48	2		n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
410	49	1		-1	30.	I	ja	Zahl[.] Leerzeichen großer Buchstabe	Gruppe 6
411	49	1		30.	r.	F	nein	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
412	49	1	1	1.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
413	49	1	2	2.	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
414	49	1	2	t.	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
415	49	1	2	t.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
416	49	2		-2	t.	N	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
417	49	2		t.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
418	49	3		-3	30.	leer	nein	Zahl[.]	Gruppe 3
419	49a	1		-1	t.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
420	49a	1		t.	m.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
421	49a	2		-2	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
422	49b	1		-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
423	49b	2		-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
424	49b	4		-4	n.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2

B.1. Manuelle Satzmusteranalyse

425	49b	4	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
426	49b	7	-7	m.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
427	49b	7	m.	m.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
428	49b	8	-8	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
429	49b	9	-9	n.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
430	49b	9	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
431	49b	10	-10	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
432	49b	11	-11	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
433	49b	12	-12	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
434	49b	13	-13	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
435	49c	1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
436	49c	2	-2	n.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
437	49c	2	n.	m.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
438	49c	3	-3	m.	B	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
439	49c	3	m.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
440	49c	4	-4	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
441	49c	5	-5	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
442	49c	6	-6	m.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
443	49c	6	m.	m.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
444	49c	7	-7	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
445	49c	8	-8	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
446	49c	9	-9	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
447	49d	1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
448	49d	2	-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
449	49d	3	-3	t.	E	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
450	49d	3	t.	2001.	leer	nein	Zahl[.]	Gruppe 3
451	49d	4	-4	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
452	49e	1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
453	49e	3	-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
454	49e	4	-4	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
455	49e	5	-5	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
456	49e	6	-6	n.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
457	49e	6	n.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
458	49e	7	-7	t.	§	ja	kleiner Buchstabe[.] Leerzeichen [§]	Gruppe 7
459	49e	7	t.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
460	49e	8	-8	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
461	49e	9	-9	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
462	49f	1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
463	49f	2	-2	d.	H	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
464	49f	2	d.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
465	49f	3	-3	t.	A	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
466	49f	3	t.	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
467	49g	1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
468	49g	2	-2	d.	leer	nein	kleiner Buchstabe[.]	Gruppe 1

B. Satzmustererkennung für Gesetzestexte

469	49g	3	-3	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
470	50		§ 50.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
471	51		§ 51.	t.	D	ja	kleiner Buchstabe[.] Leerzeichen großer Buchstabe	Gruppe 2
472	51		t.	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
473	52		§ 52.	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
474	52a		§ 52a.	e.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
475	57	1	-1	1972.	leer	nein	Zahl[.]	Gruppe 3
476	58	1	-1	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
477	58	2	-2	n.	leer	nein	kleiner Buchstabe[.]	Gruppe 1
478	58	3	-3	t.	leer	nein	kleiner Buchstabe[.]	Gruppe 4
479	58	4	-4	s.	leer	nein	kleiner Buchstabe[.]	Gruppe 1

B.2. Automatisierte Satzmusteranalyse

Die nachfolgende Tabelle zeigt die Auswertung der Tests. Die Spalte *TestXML* beschreibt das Ergebnis der Tests, dabei steht 1 für erfolgreich und 0 für fehlgeschlagen. Ergebnisse weiterer Testdurchläufe sind in der Spalte *1. Korrektur* zu sehen. Die Kommentare beschreiben grob das aufgetretene Problem.

Nr.	Paragraph	TestXML	1. Korrektur	Kommentar
1	1	0		II. Hauptstück —>Satzmarkierung nach II. Grund: Aufzählungspunkt beginnt mit einem großen Buchstaben, Satzmarkierung nach einem Punkt
2	2	1		
3	3	0	1	Beginnt ein Aufzählungspunkt mit einem kleiner Buchstaben und endet mit einem Punkt, wird ab dem ersten großen Buchstaben bis zum Ende ein Satz markiert—>Satzmarkierung, obwohl es sich um einen Teilsatz handelt —>abfangen von Aufzählungspunkten, die mit einem kleinen Buchstaben beginnen Teilsätze, die mit einem [.] enden werden teilweise als Sätze erkannt —>[.] markiert auch ein Satzende Abs. 4 wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz mit einem [.]endet. —>[.] markiert auch ein Satzende
4	4	1		
5	5	0	1	<sup>Tag im RISDOK muss beachtet werden —>hinzugefügt
6	6	0	1	Abs. 2 wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz ohne Satzzeichen ende —>Vorgängen und Nachfolger werden beachtet
7	7	1		
8	8	1		

B.2. Automatisierte Satzmusteranalyse

9	9	0	1	Abs. 2 wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz ohne Satzzeichen ende —>Vorgängen und Nachfolger werden beachtet
10	10	0	1	Abs. 5 wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz ohne Satzzeichen ende —>Vorgängen und Nachfolger werden beachtet
11	11	1		
12	12	0	1	Abs. 3: „60. Lebensjahr“ wird als Satz markiert
13	12a	1		
14	13	1		
15	14	0	1	Abs. 4: „(Anm.:“ muss beachtet werden
16	15	1		
17	15a	0		Beginnt ein Aufzählungspunkt mit einem großen Buchstaben und endet mit einem Punkt —>Satzmarkierung, obwohl es sich um einen Teilsatz handelt
18	16	0	1	Abs. 1 wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz ohne Satzzeichen endet —>Vorgängen und Nachfolger werden beachtet Abs. 7: „(Anm.:“ muss beachtet werden
19	16a	1		
20	16b	0	1	Abs. 1: gdash an anderer Stelle als bei Index 2 abfangen
21	17	1		
22	18	1		
23	18a	1		
24	18b	0	1	§ 18b.: „BGBl. Nr. 483/1984“ wird als Satz markiert
25	18c	1		
26	19	1		
27	20	0		Abs. 1 Z 2 b): Beginnt ein Aufzählungspunkt mit einem großen Buchstaben und endet mit einem Punkt —>Satzmarkierung, obwohl es sich um einen Teilsatz handelt Abs. 3 Gdash nach einem vollständigen Satz abfangen
28	21	0		
29	22	1		
30	23	1		
31	24	0	1	mehrere gdashes abfangen
32	25	1		
33	26	1		
34	27	0		Abs. 1 Z 5: Beginnt ein Aufzählungspunkt mit einem großen Buchstaben und endet mit einem Punkt —>Satzmarkierung, obwohl es sich um einen Teilsatz handelt
35	28	1		
36	29	0	1	mehrere Fehler
37	30	1		
38	31	1		
39	32	0	1	Abs. 1 Gdash nach einem vollständigen Satz abfangen
40	33	1		

B. Satzmustererkennung für Gesetzestexte

41	33a	1		
42	34	0	1	Abs. 4 und 5: „(Anm.:“ muss beachtet werden
43	34a	1		
44	35	1		
45	36	1		
46	37	0		Abs. 1 Z 1: wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz ohne Satzzeichen endet —>Vorgängen und Nachfolger werden beachtet Abs. 1: Beginnt ein Aufzählungspunkt mit einem großen Buchstaben und endet mit einem Punkt —>Satzmarkierung, obwohl es sich um einen Teilsatz handelt
47	38	0		Z 3: Beginnt ein Aufzählungspunkt mit einem großen Buchstaben und endet mit einem Punkt —>Satzmarkierung, obwohl es sich um einen Teilsatz handelt
48	39	1		
49	40	1		
50	41	1		
51	42	1		
52	42a	1		
53	43	1		
54	45	1		
55	46	1		
56	46a	0		Abs. 4: Beginnt ein Aufzählungspunkt mit einem großen Buchstaben und endet mit einem Punkt —>Satzmarkierung, obwohl es sich um einen Teilsatz handelt
57	46b	0	1	§ 46b.: <i>-Tag muss beachtet werden RegEx Exception!
58	46c			
59	47	1		
60	48	1		
61	49	1		
62	49a	1		
63	49b	0		Abs. 3: „BGBl. I Nr. 22/1997“ —>Satzmarkierung nach Nr.Abs. 3 Z 1: Beginnt ein Aufzählungspunkt mit einem großen Buchstaben und endet mit einem Punkt —>Satzmarkierung, obwohl es sich um einen Teilsatz handelt
64	49c	1		
65	49d	1		
66	49e	0	1	Abs. 2: „Abs. 4“ wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz ohne Satzzeichen endet
67	49f	1		
68	49g	1		
69	50	0	1	§50: „BGBl. Nr. 299/1979“ —>Satzmarkierung nach l.
70	51	1		
71	52	1		
72	52a	1		

B.2. Automatisierte Satzmusteranalyse

73	53	1		
74	54	1		
75	55	1		
76	56	1		
77	57	0	1	Abs. 2: „(Anm.:“ muss beachtet werden
78	58	1		
79	59	0	1	Abs. 4 wird als Satz markiert falls anschließend kein weiterer Punkt auftaucht, bspw. wenn der Teilsatz mit einem [:]endet. →[:] markiert auch ein Satzende
	Ergebnis	50	19	
	Gesamtergebnis		69	
	Prozent	0%	87,34%	

C. Anleitung zur Verwendung der Implementierung

C.1. Technische Eckdaten

Metamodelling Platform ADOxx Version	1.5
Java Version	1.8
FOP	2.0
XSLT	2.0
RIS OGD Webservice	2.2
RIS Applikation	Bundesnormen

Tabelle C.1.: Technische Eckdaten zur Implementierung

C.2. Installation

Kopieren der kompilierten Programme ins Installationsverzeichnis von ADOxx

1. Im Ordner *tools* unter "C:\Program Files\BOC\ADOxx15_EN_SA\tools\" einen Ordner *gesetzesmodelle* anlegen
2. Kopieren der folgenden Dateien in den angelegten Ordner: *searchRequest.jar*, *getDocument.jar*, *pdfExport.jar*

Import der ADOxx Bibliothek im ADOxx Development Toolkit

1. Öffnen des ADOxx Development Toolkit

2. Bereich "Library Management" auswählen
3. Unter dem Menüpunkt "Libraries" -> "Management..." auswählen
4. Im Tab "Management" den Button "Import..." auswählen
5. Im Date Explorer die Datei "SeMFIS_0.42_ADOxx_1.5_Version2.0.abl" laden

Anlegen eines Benutzers zur Bedienung der Bibliothek im ADOxx Modelling Toolkit

1. Öffnen des ADOxx Development Toolkit
2. Bereich "User Management" auswählen
3. Unter dem Menüpunkt "User" -> "User list..." auswählen
4. Button "Add" auswählen
5. Username und Passwort vergeben
6. Unter "Application library" SeMFIS_0.42_ADOxx_1.5_Version2.0 auswählen.
7. Unter "Authorisation" Modelling Toolkit auswählen.

C.3. Bedienung

C.3.1. Automatische Generierung von Gesetzesmodellen

Automatische Generierung von Gesetzesmodellen im ADOxx Modeling Toolkit

1. Mit Rechtsklick auf Ordner "Models" -> Menüpunkt "New" -> "Konfigurationsmodell" auswählen. Es wird automatisch ein Konfigurationsmodell erstellt.
2. Suche bspw. mit Kurztitel "MRG" starten. Suchergebnisse werden unter der Suchmaske aufgelistet.

3. Checkbox der Suchergebnisse auswählen.
4. Button "Visualisieren" auswählen.
5. Auswahl des Speicherorts des automatisch erstellten Gesetzesmodells.
6. Das Gesetzesmodell ist am entsprechenden Speicherort links im Modellexplorer zu finden.

C.3.2. PDF-Export

Export von Gesetzesmodellen als PDF Datei

1. Das zu exportierende Gesetzesmodell öffnen.
2. Bereich "Import/Export" auswählen.
3. Menüpunkt "Model" -> "PDF Export" auswählen.
4. Speicherort der Datei auswählen und Dateinamen eingeben.
5. Auf "Speichern" klicken.
6. Die PDF-Datei ist am angegebenen Speicherort zu finden.

C.4. Implementierung

Die Implementierung ist in Java realisiert. Die Quellen der Java Dateien sind der Arbeit beigelegt. Die Programme sind zudem im ausführbarem Format (.jar) vorzufinden. Diese .jar-Dateien werden über ADOScript im Tool aufgerufen und ausgeführt.

C.4.1. Suche nach Gesetzestexten

Bei *searchRequest.jar* handelt es sich um die kompilierte Datei für die Suche nach Gesetzestexten. Als Argumente sind folgende zwei Variablen mitzugeben:

C. Anleitung zur Verwendung der Implementierung

- *suche* - kommagetrennte Strings, welche in folgender Reihenfolge die Suchbegriffe darstellen: Suchworte, Titel/Abkürzung, Paragraph von, Paragraph bis, Artikel von, Artikel bis und Fassung vom
- *dateipfad* - ein Dateipfad zum Speichern der Suchergebnisse

```
1 SET suche:(suchworte + "," + titelAbk + "," + paragraphVon + "," +  
2 paragraphBis + "," + artikelVon + "," + artikelBis + "," + fassungVom)  
3  
4 SYSTEM ("cmd /c java -jar \"\" + path_only +  
5 \"\\tools\\gesetzesmodelle\\searchRequest.jar\" \"\" + suche +  
6 \"\" \"\" + dateipfad + "\" true > C:\\Temp\\searchRequest_log.txt")
```

Abbildung C.1.: Aufruf des Programms zur Suche von Gesetzestexten

C.4.2. Visualisierung von Gesetzestexten

Bei *importLaws.jar* handelt es sich um die kompilierte Datei für die Visualisierung von Gesetzestexten. Als Argumente sind folgende zwei Variablen mitzugeben:

- *dokumentenummer* - die Dokumentenummer des entsprechenden Gesetzestextes/-paragraphen
- *dateipfad* - der Dateipfad zum Speichern der Datei

```
1 SYSTEM ("cmd /c java -jar \"\" + path_only +  
2 \"\\tools\\gesetzesmodelle\\getDocument.jar\" \"\" + dokumentenummer +  
3 \"\" \"\" + dateipfad + "\" true > C:\\Temp\\importGesetzesmodell_log.txt")
```

Abbildung C.2.: Aufruf des Programms zur Visualisierung eines Gesetzesmodells

C.4.3. Export von Gesetzestexten als PDF

Bei *pdfExport.jar* handelt es sich um die kompilierte Datei für den PDF-Export von Gesetzestexten. Als Argumente sind folgende drei Variablen mitzugeben:

- *adoxmlfile* - der Dateipfad zur ADOXML-Datei mit dem Gesetzesmodell
- *pdffile* - der Dateipfad zum Speichern der PDF-Datei
- *xslfile* - der Dateipfad zum XSL-Stylesheet

```
1 SYSTEM ("cmd /c java -jar \"\" + path_only +  
2 \"\\tools\\gesetzesmodelle\\pdfExport.jar\" \"\" + tempadoxmlfile + \"\"  
3 \"\" + tempdffile + \"\" \"\" + tempxslfile + \"\" true > C:\\Temp\\PDFExport_log.txt")
```

Abbildung C.3.: Aufruf des Programms zum PDF-Export eines Gesetzesmodells

D. Relevante Code-Ausschnitte

D.1. ADOScript

```
1 #Konfigurationsoberfläche
2 CC "Core" GET_CLASS_ID classname:("Konfiguration")
3 SET konfigClassId:(classid)[...]
4 CC "Core" GET_MODEL_INFO modelid:(actModelId)
5 SETL sCreatedModelType:(modeltype)
6 IF (sCreatedModelType = "Konfigurationsmodell") {
7     CC "Core" CREATE_OBJ modelid:(actModelId) classid:(konfigClassId)
8     objname:""
9     SETL konfigObjId:(objid)
10    CC "Core" SET_ATTR_VAL objid:(konfigObjId) attrname:("Position")
11    val:"NODE x:2.5cm y:1cm w:20cm h:9.5cm index:1"
12    CC "Core" SET_ATTR_VAL objid:(konfigObjId) attrname:("Name")
13    val:"Suchmaske, um Gesetzestexte abzufragen und anschliessend zu
14    visualisieren."
15    CC "Modeling" SET_MOUSE_ACCESS objid:(konfigObjId) access:0
16    #Suchworte
17    CC "Core" CREATE_OBJ modelid:(actModelId) classid:(suchworteClassId)
18    objname:""
19    SETL suchworteObjId:(objid)
20    CC "Core" SET_ATTR_VAL objid:(suchworteObjId) attrname:("Position")
21    val:"NODE x:7cm y:4cm w:9cm h:1.2cm index:1"[...]
22 }
```

Tabelle D.1.: Erstellung einer Suchmaske im Tool(ADOxx)

D.2. OGD RIS Webservice

```
1 public static void main(String args[]) throws
2 ParserConfigurationException, SAXException, IOException,
3 TransformerException, XMLStreamException, JAXBException {
4     String docId = (args.length > 1)? args[0] : "NOR40021888";
5     String fileName = (args.length > 1) ? args[1] : "adoxx.xml";
6     try {
7         // Create SOAP Connection
8         SOAPConnectionFactory soapConnectionFactory =
9         SOAPConnectionFactory.newInstance();
10        SOAPConnection soapConnection =
11        soapConnectionFactory.createConnection();
12        /** Send SOAP Message to SOAP Server, neue Version 2.2
13        ab 01.07.2016 */
14        String url = "http://data.bka.gv.at/ris/v2_2/ogdrisservice.asmx";
15        SOAPRequest soapRequest = new SOAPRequest();
16        SOAPMessage soapRequestMessage =
17        soapRequest.addDocument(docId);
18        SOAPMessage soapResponse =
19        soapConnection.call(soapRequestMessage, url);
20        [...]
21        // Process the SOAP Response
22        String resultString = printSOAPResponse(soapResponse);
23        SOAPResult result = new SOAPResult(resultString);
24        result.createResult();
25        result.saveDocumentResults(fileName);
26        soapConnection.close();
27    } catch (Exception e) {
28        System.err.println("Error occurred while sending SOAP Request to Server");
29        e.printStackTrace();
30    }
31 }
```

Tabelle D.2.: SOAP Dokumentenabfrage über OGD RIS Webservice 2.2

```
1 public class SOAPClientRequest {
2     public static void main(String args[]) { [...] //Setzen der Variablen
3         try {
4             // Create SOAP Connection
5             SOAPConnectionFactory soapConnectionFactory =
6             SOAPConnectionFactory.newInstance();
7             SOAPConnection soapConnection =
8             soapConnectionFactory.createConnection();
9
10            /** Send SOAP Message to SOAP Server, neue Version 2.2
11            ab 01.07.2016 */
12            String url = "http://data.bka.gv.at/ris/v2_2/ogdrisservice.asmx";
13            SOAPRequest soapRequest = new SOAPRequest();
14            SOAPMessage soapRequestMessage =
15            soapRequest.addRequest(suchworte, titelAbk, paragraphVon,
16            paragraphBis, artikelVon, artikelBis, fassungVom);
17            [...]
18            soapRequestMessage.saveChanges();
19            SOAPMessage soapResponse =
20            soapConnection.call(soapRequestMessage, url);
21
22            // Process the SOAP Response
23            String responseString = printSOAPResponse(soapResponse);
24            SOAPResult result = new SOAPResult(responseString);
25            result.createResult();
26            result.saveSearchResults(fileName);
27            soapConnection.close();
28        } catch (Exception e) {
29            System.err.println("Error occurred while sending SOAP Request to Server");
30            e.printStackTrace();
31        }
32    }[...]
33 }
```

Tabelle D.3.: SOAP Suchabfrage über OGD RIS Webservice 2.2

Literaturverzeichnis

- Alpar, P. und S. Olbrich (2005). "Legal Requirements and Modelling of Processes in e- Government". In: *Electronic Journal of e-Government* 3.3, S. 107 –116.
- Awad, A., G. Decker und M. Weske (2008). "Efficient Compliance Checking Using BPMN-Q and Temporal Logic". In: *Business Process Management: 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings*. Ed. by M. Dumas, M. Reichert und M.-C. Shan. Springer Berlin Heidelberg, S. 326 –341.
- Awad, A. und M. Weske (2010). "Visualization of Compliance Violation in Business Process Models". In: *Business Process Management Workshops: BPM 2009 International Workshops, Ulm, Germany, September 7, 2009. Revised Papers*. Ed. by S. Rinderle-Ma, S. Sadiq und F. Leymann. Springer Berlin Heidelberg, S. 182–193.
- Bergmans, B. (2009). *Visualisierung in Rechtslehre und Rechtswissenschaft: Ein Beitrag zur Rechtsvisualisierung*. Logos Verlag Berlin GmbH.
- Bork, D. und H.-G. Fill (2014). "Formal Aspects of Enterprise Modeling Methods: A Comparison Framework". In: *47th Hawaii International Conference on System Sciences (HICSS)*. IEEE Computer Society, S. 3400–3409.
- Bundeskanzleramt der Republik Österreich (1990). *Handbuch der Rechtssetzungstechnik: Teil 1: Legistische Richtlinien 1990*. URL: http://www.konvent.gv.at/K/DE/INST-K/INST-K_00044/fname_040076.pdf (visited on 09/15/2016).
- Bundeskanzleramt der Republik Österreich (2016a). *Cooperation OGD Österreich*. URL: <https://www.data.gv.at/infos/cooperation-ogd-oesterreich/> (visited on 09/21/2016).

- Bundeskanzleramt der Republik Österreich (2016b). *OgdRisService Web Service*.
URL: http://data.bka.gv.at/ris/v2_2/ogdrisservice.aspx (visited on 03/12/2017).
- Bundeskanzleramt der Republik Österreich (2016c). *RIS - Allgemeine Informationen*.
URL: <https://www.ris.bka.gv.at/UI/Info.aspx> (visited on 09/21/2016).
- Bundeskanzleramt der Republik Österreich (2016d). *RIS - Open Government Data*.
URL: <https://www.ris.bka.gv.at/UI/Ogd.aspx> (visited on 09/21/2016).
- Card, S. K., J. D. Mackinlay und B. Shneiderman (1999). "Using vision to think".
In: *Readings in information visualization*, S. 579–581.
- Das Österreichische Parlament (2017a). *Das elektronische Gesetzgebungsverfahren*.
URL: <https://www.parlament.gv.at/PERK/GES/ERECHT/index.shtml> (visited on 09/28/2016).
- Das Österreichische Parlament (2017b). *Der Weg eines Bundesgesetzes*. URL: https://www.parlament.gv.at/ZUSD/PDF/Weg_der_Bundesgesetzgebung.pdf (visited on 05/02/2017).
- Das Österreichische Parlament (2017c). *Gesetzesinitiativen*. URL: <https://www.parlament.gv.at/PERK/GES/WEG/INITIATIVE/index.shtml> (visited on 05/02/2017).
- Engesser, M., ed. (2014). *UML und Sprache – Wirklichkeit und Technik*. Jusletter IT.
- Enser, G., G. Quirchmayr, R. Traunmüller und N. Wilfert (1989). "Der Einsatz von Expertensystemtechniken zur Unterstützung der Arbeit mit Rechtsinformationssystemen". In: *Paul M. (eds) GI — 19. Jahrestagung II. Informatik-Fachberichte*. Vol. 223. Springer, Berlin, Heidelberg, S. 125–141.
- Falleri, J.-R., M. Huchard, M. Lafourcade und C. Nebut (2008). "Metamodel Matching for Automatic Model Transformation Generation". In: *MoDELS, vol. 5301 of Lecture Notes in Computer Science*. Ed. by K. Czarnecki, I. Ober, J.-M. Bruel, A. Uhl und M. Völter. Springer, S. 326–340.
- Fill, H.-G. (2009). *Visualisation for Semantic Information Systems*. Gabler.
- Fill, H.-G. (2015). "Metamodeling as an Interface between Syntax and Semantics". In: *Elektronische Schnittstellen in der Staatsorganisation - Festschrift zum 60.*

- Geburtstag von Josef Souhrada*. Ed. by B. Glück, F. Lachmayer, G. Schefbeck und E. Schweighofer. Books@OCG, S. 43–50.
- Fill, H.-G. und A. Grieb (2017). “Visuelle Modellierung des Rechts: Vorgehensweise und Praktische Umsetzung für Rechtsexperten”. In: *Conference Proceedings IRIS 2017*. Ed. by E. Schweighofer. Weblaw.
- Fill, H.-G. und D. Karagiannis (2013). “On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform”. In: *Enterprise Modelling and Information Systems Architectures - An International Journal*. Vol. 8, S. 4–25.
- Forschungsgruppe für Industrielle Software (INSO) (2008). *Informationstechnische und operational-strategische Überlegungen zur Zukunft der elektronischen Unterstützung der Rechtssetzung*. URL: <http://www.univie.ac.at/RI/IRIS2008/papers/grechenig.pdf> (visited on 03/26/2017).
- Hahn, T., B. Mielke und C. Wolff, eds. (2013). *Juristische Lehrcomics – Anforderungen und Möglichkeiten*. Jusletter IT.
- Houy, C., P. Fettke, P. Loos, I. Speiser, M. Herberger, A. Gass und U. Nortmann (2012). “ARGUMENTUM – Towards Computer-Supported Analysis, Retrieval and Synthesis of Argumentation Structures in Humanities Using the Example of Jurisprudence”. In: *Poster and Demo Track of the 35th German Conference on Artificial Intelligence (KI 2012)*.
- Houy, C., T. Niesen, J. Calvillo, P. Fettke, P. Loos, A. Krämer und et al. (2015). “Konzeption und Implementierung eines Werkzeuges zur automatisierten Identifikation und Analyse von Argumentationsstrukturen anhand der Entscheidungen des Bundesverfassungsgerichts im Digital-Humanities-Projekt ARGUMENTUM”. In: *Datenbank Spektrum*, S. 1–9.
- Hudec, M. (2007). *Statistik 1 für SoziologInnen: Einfach Häufigkeitstabellen und Grundprinzipien der Visualisierung von Quantitativen Daten*. URL: <http://homepage.univie.ac.at/marcus.hudec/Lehre/SS2007/Statistik1/p2C2.pdf> (visited on 04/23/2017).
- Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart (1995). *STTS Tag Table (1995/1999)*. URL: <http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html> (visited on 05/10/2017).

- Ismayr, W. (2008a). "Gesetzgebung in den Staaten der Europäischen Union im Vergleich". In: *Gesetzgebung in Westeuropa : Eu-Staaten und Europäische Union*. Wiesbaden : VS, Verl. für Sozialwiss.: Ismayr, Wolfgang, S. 9 –64.
- Ismayr, W. (2008b). *Gesetzgebung in Westeuropa : Eu-Staaten und Europäische Union*. Wiesbaden : VS, Verl. für Sozialwiss.: Ismayr, Wolfgang.
- Jandach, T. (1993). *Juristische Expertensysteme: Methodische Grundlagen ihrer Entwicklung*. Springer-Verlag Berlin, Heidelberg.
- Jurafsky, D. und J. Martin (2000). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall: Prentice Hall series in artificial intelligence.
- Kahlig, W., ed. (2008). *UML für juristische Anwendungen*. Internationales Rechtsinformatik Symposium (IRIS). Jusletter IT.
- Visualisierungstypologie des Deutschen Privatrechts* (2011). Jusletter IT.
- Kahlig, W. und E. Kahlig (2013). "Legistical Efforts By E-Government". In: *The Third International Conference on Digital Information Processing and Communications*, S. 391–403.
- Karagiannis, D. und H. Kühn (2002). *Metamodelling Platforms*. Springer-Verlag.
- Knackstedt, R., C. Brelage und N. C. Kaufmann (2006). "Entwicklung rechtssicherer Web-Anwendungen - Strukturierungsansatz, State-of-the-Art und ausgewählte Aspekte der fachkonzeptionellen Modellierung". In: *Wirtschaftsinformatik* 48.1, S. 27 –35.
- Kölbl, C. (2015). "Ein Recht auf verständliche Rechtstexte/-normen?" In: *juridikum* 3, S. 373.
- Koller, D. und N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)*. The MIT Press.
- Kordjamshidi, P. (2013). "Structured Machine Learning for Mapping Natural Language to Spatial Ontologies". PhD thesis. KU Leuven – Faculty of Engineering.
- Kowalski, C. (1987). *Lösungsansätze für juristische Expertensysteme*.
- Kramer, J. (2007). "Is abstraction the key to computing?" In: *Communications of the ACM* 50.4, S. 36–42.

- Lachmayer, K. (2015). "Legistik zwischen Wissen und politischer Entscheidungsfindung". In: *juridikum* 3, S. 333.
- Manning, C. D., P. Raghavan und H. Schütze (2009). *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press.
- Manning, C. D. und H. Schütze (1999). *Foundations of statistical natural language processing*. Vol. 999. MIT Press.
- Mazza, R. (2009). *Introduction to Information Visualization*. Springer-Verlag Berlin, Heidelberg.
- Merkel, D., E. Schweighofer und W. Winiwarter (1994). "CONCAT — Connotation analysis of thesauri based on the interpretation of context meaning". In: *Database and Expert Systems Applications: 5th International Conference, DEXA '94 Athens, Greece, September 7–9, 1994 Proceedings*. Ed. by D. Karagiannis. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 329–338.
- Mertens, P., F. Bodendorf, W. König, A. Picot, M. Schumann und T. Hess (2012). *Grundzüge der Wirtschaftsinformatik*. Springer-Verlag.
- Nabizai, A. und H.-G. Fill (2017). "Eine Modellierungsmethode zur Visualisierung und Analyse von Gesetzestexten". In: *Jusletter IT*, 23. Februar 2017. Weblaw.
- Olbrich, S. (2007). "Visualisierung von rechtlichen Rahmenbedingungen in Geschäftsprozessmodellen". In: *10 Jahre IRIS: Bilanz und Ausblick*. Ed. by E. Schweighofer, A. Geist und G. Heindl. Richard Boorberg Verlag, S. 474–479.
- OMG (2009). *OMG Unified Modeling Language™ (OMG UML), Superstructure*.
- Österle, H., J. Becker, U. Frank, T. Hess, D. Karagiannis, H. Krcmar, P. Loos, P. Mertens, A. Oberweis und E. J. Sinz (2011). "Memorandum on design-oriented information systems research". In: *European Journal of Information Systems* 20, S. 7–10.
- Palmer, D. D. und M. A. Hearst (1997). "Adaptive Multilingual Sentence Boundary Disambiguation". In: *Comput. Linguist.* 23.2, S. 241–267.
- Parycek, P. und J. Höchtl, eds. (2014). *Open Government Data in Österreich*. Jusletter IT.

- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publisher, Inc.
- Pelinka, A. (2008). "Gesetzgebung im politischen System Österreichs". In: *Gesetzgebung in Westeuropa : Eu-Staaten und Europäische Union*. Wiesbaden : VS, Verl. für Sozialwiss.: Ismayr, Wolfgang, S. 431 –461.
- Pelinka, A. und S. Rosenberger (2003). *Österreichische Politik: Grundlagen, Strukturen, Trends*. Pelinka, Anton and Rosenberger, Sieglinde.
- Reimer, S., ed. (2008). *Rechtssetzung zwischen Österreich und Tasmanien*. Internationales Rechtsinformatik Symposium (IRIS). Jusletter IT.
- Reynar, J. C. und A. Ratnaparkhi (1997). "A Maximum Entropy Approach to Identifying Sentence Boundaries". In: *Proceedings of the Fifth Conference on Applied Natural Language Processing*. Washington, DC: Association for Computational Linguistics, S. 16–19.
- Rinderle-Ma, S., L. T. Ly und P. Dadam (2008). "Aktuelles Schlagwort: Business Process Compliance". In: *EMISA Forum*, S. 24 –29.
- Saravanan, M., B. Ravindran und S. Raman (2006). "Improving Legal Document Summarization Using Graphical Models". In: *Legal Knowledge and Information Systems: JURIX 2006: The Nineteenth Annual Conference*. Ed. by T. M. van Engers. IOS Press, Netherlands, S. 51 –61.
- Sauerwald, M. J. (2007). "Mind Mapping als Visualisierungsmethode in der juristischen Praxis". In: *10 Jahre IRIS: Bilanz und Ausblick*. Ed. by E Schweighofer, A. Geist und G. Heindl. Internationales Rechtsinformatik Symposium (IRIS). Richard Boorberg Verlag, S. 480–485.
- Scheer, A.-W., H. Krupke und R. Heib (2003). *E-Government - Prozessoptimierung in der öffentlichen Verwaltung*. Berlin Heidelberg: Springer Verlag.
- Schefbeck, G. (2007). "E-Partizipation und elektronisches Begutachtungsverfahren". In: *Bildungsprotokolle - 5. Klagenfurter Legistik Gespräche 2007*. Klagenfurt: Kärntner Verwaltungsakademie, S. 7 –29.
- Schleicher, D., M. Weidmann, F. Leymann und D. Schumm (2010). "Compliance Scopes: Extending the BPMN 2.0 Meta Model to Specify Compliance Require-

- ments". In: *Proceedings of SOCA 2010*. Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany. IEEE Computer Society.
- Schneider, E. und F. E. Schnapp (2006). *Logik für Juristen*. München: Verlag Vahlen.
- Schoormann, T. und R. Knackstedt, eds. (2016). *Rechtsvisualisierung mit ComicTools – Vergleich und Weiterentwicklungsperspektiven*. Jusletter IT.
- Schweighofer, E. (1999). *Legal Knowledge Representation: Automatic Text Analysis in Public International and European Law*. Kluwer Law Intl.
- Schweighofer, E. und W. Winiwarter (1993). "Legal Expert System KONTERM - Automatic Representation of Document Structure and Contents". In: *Proceedings of the 4th International Conference on Database and Expert Systems Applications*. DDEXA '93. London, UK, UK: Springer-Verlag, S. 486–497.
- Schweighofer, E., W. Winiwarter und D. Merkl (1995). "Information Filtering: The Computation of Similarities in Large Corpora of Legal Texts". In: *Proceedings of the 5th International Conference on Artificial Intelligence and Law*. ICAIL '95. New York, NY, USA: ACM, S. 119–126.
- Soltana, G., E. Fourneret, M. Adedjouma, M. Sabetzadeh und L. Briand (2014). "Using UML for Modeling Procedural Legal Rules: Approach and a Study of Luxembourg's Tax Law". In: *Model-Driven Engineering Languages and Systems 8767*, S. 450–466.
- Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Springer.
- Stöger, H. und H. Weichsel, eds. (2008). *Das Redesign des Rechtsinformationssystems - RIS*. Vol. 11. Internationales Rechtsinformatik Symposium (IRIS). Jusletter IT.
- Strahonja, V. (2006). "Modeling legislation by using UML state machine diagrams". In: *Electrical and Computer Engineering*. IEEE, S. 624–627.
- Tobler, C. und J. Beglinger (2010). *Essential EU Law in Charts*. HVG-ORAC, Budapest.
- Toutanova, K., D. Klein, C. D. Manning und Y. Singer (2003). "Feature-rich part-of-speech tagging with a cyclic dependency network". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Compu-*

- tational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, S. 173–180.
- van Engers, T. M., R. Gerrits, M. Boekenoogen, E. Glassée und P. Kordelaar (2001). "POWER: Using UML/OCL for Modeling Legislation - an Application Report". In: *Proceedings of the 8th International Conference on Artificial Intelligence and Law*. New York, NY, USA: ACM, S. 157–167.
- Van Gog, R. und T. M. Van Engers (2001). "Modeling legislation using natural language processing". In: *Systems, Man, and Cybernetics, 2001 IEEE International Conference*. Vol. 1. IEEE.
- Walker, D. J., D. E. Clements, M. Darwin und J. W. Amtrup (2001). "Sentence boundary detection: A comparison of paradigms for improving MT quality". In: *In Proceedings of MT Summit VIII: Santiago de Compostela*, S. 18–22.
- Walser Kessel, C., F. Lachmayer, V. Čyras, P. Parycek und Y.-H. Weng, eds. (2016). *Rechtsvisualisierung als Vernetzung von Sprache und Bild – Anmerkungen zum Buch «Kennst du das Recht?»* Jusletter IT.
- Walter, S. (2009). "Definition extraction from court decisions using computational linguistic technology". In: *Formal Linguistics and Law*. Ed. by G. Grewendorf und M. Rathert, S. 183 –223.
- Weichsel, H., ed. (2014). *Rechtsinformationssystem (RIS) – ein Rück- und Ausblick*. Jusletter IT.
- Wilde, T. und T. Hess. "Forschungsmethoden der Wirtschaftsinformatik - Eine empirische Untersuchung". In: *Wirtschaftsinformatik 49*, S. 280–287.
- Wyner, A., R. Mochales-Palau, M.-F. Moens und D. Milward (2010). "Approaches to Text Mining Arguments from Legal Cases". In: *Semantic Processing of Legal Texts*. Springer-Verlag Berlin, Heidelberg, S. 60–79.