

DIGITALISIERUNG DER GESETZGEBUNG ZUR STEIGERUNG DER DIGITALEN SOUVERÄNITÄT DES STAATES

Bernhard Rumpe
Judith Michael
Oliver Kautz
Roland Krebs
Sabine Gandenberger
Janos Standt
Uli Weber



[RMK+21] B. Rumpe, J. Michael, O. Kautz, R. Krebs, S. Gandenberger, J. Standt, U. Weber:
Digitalisierung der Gesetzgebung zur Steigerung der digitalen Souveränität des Staates.
In: Berichte des NEGZ, 19, Nationales E-Government Kompetenzzentrum e. V., June 2021.
www.se-rwth.de/publications/

Für einen modernen Staat

Das Nationale E-Government Kompetenzzentrum vernetzt Experten aus Politik, Verwaltung, Wissenschaft und Wirtschaft und ist die zentrale, unabhängige Plattform für Staatsmodernisierung und Verwaltungstransformation in Deutschland.

Herausgegeben und gefördert vom
Nationalen E-Government Kompetenzzentrum e. V.
Berlin 2021

INHALT

Zusammenfassende Empfehlungen	5
1. Motivation und Ziele der Kurzstudie	6
1.1 Problemstellung	6
1.2 Lösungsansatz	7
1.3 Aufbau der Kurzstudie	7
2. Wissenschaftlicher und praktischer Hintergrund	8
2.1 Aktuelle Diskussion in Praxis und Wissenschaft	8
2.2 Wichtige IT Begriffe und Prinzipien	10
2.3 DSLs für Gesetze und Verträge	11
2.4 Ontologien für die Gesetzesdomäne	13
2.5 Erfahrungen aus der Praxis: ELSTER	14
2.6 Juristische Rahmenbedingungen für die Digitalisierung der Gesetzgebung und von Gesetzen	16
2.6.1 Digitalisierungsgrad im Steuerrecht	16
2.6.2 Das Gesetzgebungsverfahren	17
2.6.3 Die juristische Gesetzesauslegung als Basis für eine DSL	21
2.6.4 Vom Kodex-Gesetz zur DSL und Modellen	22
2.6.5 Verhältnis von juristischen Gesetzestexten zu DSLs bzw. Modellen	23
3. Ergebnisse: Lösungsansätze für Steuergesetze und Diskussion möglicher Varianten	23
3.1 Mögliche Lösungsvarianten	24
3.1.1 Charakteristik von Gesetzestexten	24
3.1.2 Der Zusammenhang zwischen Gesetzestexten und Modellen	25
3.1.3 Zeitpunkt der Erstellung der Modelle	26
3.1.4 Prozessschritte und Nutzergruppen	26
3.1.5 Tooling und Interpretation	28
3.2 Eine DSL zur Abbildung von gesetzlichen Steuerberechnungen und Validierung	28
3.2.1 Felder-Modell	28
3.2.2 Aufbau der Sprache	31
3.2.3 Kurzüberblick über vorhandene Operatoren	32
3.2.4 Notwendige Erweiterungen	32
3.3 Eine DSL zur Modellierung von gesetzlichen Fristen	33
3.3.1 Festsetzungsverjährung, Fristen und Termine im Steuerrecht	34
3.3.2 Datenmodell zur Berechnung von Fristen	35
3.3.3 Interpretation auf Basis logischer Formeln	36
3.3.4 Interpretation auf Basis von Bedingungen, Zuweisungen und Anweisungen	39
3.3.5 Modellierung von Gesetzestexten zur Beschreibung von Fristen	40
3.3.6 Codegenerierung aus Gesetzestexten zur Beschreibung von Fristen	45
3.4 Zusammenspiel von DSLs und unserem Ansatz	47

4. Handlungsempfehlungen für die Umsetzung	50
5. Zusammenfassung und Ausblick	52
Literatur	53
Über die Autorinnen und Autoren	57
Kommentar	59
Impressum	60

ZUSAMMENFASSENDE EMPFEHLUNGEN

In dieser Studie wird gezeigt, dass die Darstellung von Gesetzen durch Domänen-spezifische Sprachen (eng. Domain Specific Languages, DSLs) und die automatisierte Auswertung der mit der DSL formulierten Modelle möglich ist. Eine DSL ist eine formale Sprache (im Gegensatz zur natürlichen Sprache), die für ein bestimmtes Problemfeld, die sogenannte Domäne, entworfen und entwickelt wird. Wir zeigen die praktische Realisierbarkeit an Beispielen aus dem Steuerrecht, die Anwendung ist jedoch auch für Gesetzestexte aus unterschiedlichen Domänen z.B. Soziales, Umwelt oder Strafrecht durchführbar. Im Strafrecht könnte zur Begründungsunterstützung bspw. eine DSL entwickelt werden, die konkrete Ermessensspielräume als Intervalle, gegebenenfalls ergänzt um KI-basierte Durchschnittswerte, bei ähnlichen Fällen berechnet.

Es ist daher an der Zeit konkrete Projekte zu definieren und durchzuführen, in denen praktisch einsetzbare Werkzeuge entstehen. Konkret schlagen wir die folgenden Umsetzungsschritte vor:

- 1. Bildung interdisziplinärer Arbeitsgruppen zur Entwicklung von DSLs zur Abbildung von Gesetzestexten (organisatorisch und technisch):** Durch die Verwendung von DSLs wird die Digitalisierung von Gesetzen in vielen Bereichen erst ermöglicht bzw. in anderen beschleunigt, da sich – soweit möglich – eine objektiv richtige Gesetzesanwendung direkt in Software übersetzen lässt. Zur Entwicklung von DSLs für die Formulierung von unterschiedlichen Gesetzestexten ist das Wissen von unterschiedlichen Beteiligten notwendig wie z.B. Juristen, Verwaltungsexperten, und Informatikern. Diese Zusammensetzung muss sich für unterschiedliche Domänen (z.B. Steuern, Soziales, Umwelt) unterscheiden, da die jeweilige fachliche Expertise benötigt wird.
- 2. Entwicklung der Werkzeuge zur Verarbeitung und Code-Generierung aus Modellen (inhaltlich und technisch):** Für den praktischen Einsatz der DSLs müssen sich die rele-

vanten Nutzergruppen darauf verständigen, welche IT Werkzeuge zu einem Gesetzestext notwendig sind. In einem weiteren Schritt sind diese Werkzeuge umzusetzen.

- 3. Definition von DSLs und Modellen im Gesetzgebungsprozess, Generierung der natürlichsprachlichen Gesetzesfassungen und Generierung von Werkzeugen aus Modellen zur Unterstützung der Gesetzesanwendung (inhaltlich und technisch):** Diese Handlungsempfehlung zielt direkt auf die Steigerung der digitalen Souveränität des Staats ab, da die Gesetzgebung die Möglichkeit haben muss, möglichst präzise Gesetze formulieren zu können. Unter bestimmten Voraussetzungen plädieren die Studienautoren für die Verabschiedung von Modellen im Rahmen von Gesetzgebungsverfahren. Wir empfehlen konkrete Projekte umzusetzen. Dies kann die Arbeit an neuen Gesetzen, weiterführend aber auch an bestehenden Gesetzen sein.
- 4. Weiterbildung in der Verwaltung, von Juristen und politischen Entscheidungsträgern, um die Formulierung von Gesetzen in Modellen und deren Verständnis zu ermöglichen:** Die Beschreibung von Gesetzen in Modellen macht es notwendig auch auf juristischer und politischer Ebene entsprechende Kompetenzen aufzubauen.
- 5. Anhebung von DSLs auf die Ebene von Standards und Normen:** DSLs, Modelle und Berechnungsvorschriften werden als Standards und Normen definiert, die sowohl in der zuständigen Behörde, als auch in der unternehmensinternen Software direkt zur Ausführung kommen können und daher zu einer wesentlich effizienteren IT und auch Zusammenarbeit der IT einzelner Verwaltungen und Unternehmen beitragen. Sie stellen damit einen essentiellen Teil der Digitalisierung dar.

Schlagworte: Digitale Souveränität, Gesetzgebung, Gesetze, Modelle, Domänenspezifische Sprache

1. MOTIVATION UND ZIELE DER KURZSTUDIE

1.1 Problemstellung

Insbesondere im Bereich des öffentlichen Rechts, welches das Verhältnis zwischen Trägern der öffentlichen Gewalt (Staat) und einzelnen Privatrechtssubjekten (Bürgern) regelt, sollte es höchste Priorität haben, dass Gesetze so formuliert, konkretisiert und ausgestaltet werden, dass diese keinen Raum für fehlerhafte Interpretation und Anwendung durch die Verwaltungen oder andere Träger der öffentlichen Gewalt geben, indem durch die Formulierung von möglichst eindeutigen und klaren Regelungen präzise Gesetze geschaffen werden. Mittelbar könnte somit ebenfalls vermieden werden, dass Verwaltungsverlautbarungen wie Erlasse oder Richtlinien veröffentlicht werden, die nicht im Einklang mit dem Gesetz stehen.¹ Soweit möglich, soll eine objektiv richtige Gesetzesanwendung gewährleistet werden.

Mit zunehmender Prozess-Automatisierung und -Digitalisierung, die auch in die Arbeitsbereiche der Verwaltungen Einzug erhält, gewinnt dieser Aspekt an Brisanz, den die Väter des Grundgesetzes unmöglich bedacht haben konnten. Denn mit dem Einsatz von automationsgestützten Systemen in der Verwaltung, müssen abgeleitete Gesetzesanwendungen durch die Verwaltung und deren Dienstleister programmiert und dazu *Gesetzestexte in eindeutiger, maschinenlesbarer Form* ausgedrückt werden, um sie in Software umsetzen zu können. Diese Übersetzung von Gesetzestext in Programmcode

birgt das Risiko einer weiteren **Abweichung bzw. Entfernung** von der durch den Gesetzgeber gewollten Gesetzesregelung (siehe Abb. 1). Nach Auffassung der Studienautoren ist vor diesem Hintergrund besonderes Augenmerk auf die Schaffung von Voraussetzungen für eine rechtssichere und praktisch auch durchzuführende Gesetzesanwendung durch die Exekutive zu legen.

Nach Auffassung der Studienautoren sollte aus den genannten Gründen sichergestellt werden, dass insbesondere Politikern und Nicht-Steuerexperten, die überwiegend am Gesetzgebungsprozess, nämlich bei Gesetzesinitiativen, der Arbeit in Ausschüssen, in verschiedenen Lesungen und schließlich bei der Verabschiedung von Gesetzen beteiligt sind und damit die maßgeblichen Entscheidungen über Gesetze treffen, ein Instrument an die Hand gegeben wird, das es ihnen ermöglicht, die gewollte Gesetzesanwendung durch die Verwaltung sicherzustellen und das Risiko einer fehlerhaften Gesetzesanwendung mittels IT zu reduzieren. Unser Ansatz steigert demnach die digitale Souveränität des Staates, indem der Wille des Gesetzgebers präziser zum Ausdruck gebracht werden kann.

Ein Abweichen von der intendierten Gesetzesanwendung kann gerade durch den Einsatz von automationsgestützten Systemen der Verwaltung vermieden werden. Ein solches Instrument könnte dabei oftmals initial durch die fachlichen Experten der Verwaltung selbst eingesetzt wer-



Abbildung 1: Abweichungen von der vom Gesetzgeber intendierten Gesetzesanwendung.

¹ Ein Auseinanderfallen der Gesetzesinterpretation zwischen Steuerpflichtigen und der Finanzverwaltung auf Basis von Verwaltungsverlautbarungen ist regelmäßig Gegenstand von Finanzgerichtsentscheidungen.

den, die in der Praxis – beispielsweise im Auftrag der Bundesregierung – Gesetzesvorlagen erarbeiten. Ggf. kann so auch der rechtssichere Erlass von Verwaltungsanweisungen, die im Einklang mit dem Willen des Gesetzgebers stehen, unterstützt werden und womöglich gar eine Reduktion von außergerichtlichen und gerichtlichen Rechtsbehelfsverfahren nach sich ziehen.

Im direkten Vergleich zum dargelegten Vorschlag ist festzustellen, dass der Gesetzgeber bei der Verabschiedung von Gesetzen aktuell wenig bis gar nicht auf deren konkrete *digitale Umsetzung* achtet.²

Da die Prozess-Automatisierung und Digitalisierung gerade in der Finanzverwaltung relativ weit vorangeschritten ist (z.B. vollautomatische Verwaltungsakte im Steuerverfahren (Kar et al. 2019) – vielen Steuerpflichtigen in Deutschland dürfte das bereits 1996 gestartete Projekt „Elektronische Steuererklärung (ELSTER)“ für die digitale Abgabe der Steuererklärung bekannt sein – eignet sich gerade das Rechtsgebiet des Steuerrechts, um im Rahmen dieser Studie genauer betrachtet zu werden. Im Folgenden wird daher, wo es zweckmäßig erscheint, auf die Steuergesetzgebung, die Steuergesetze und den Steuerverwaltungsapparat Bezug genommen.

1.2 Lösungsansatz

Vorbehaltlich der Durchführungsmöglichkeit, insbesondere der verfassungsrechtlichen Zulässigkeit, die ggf. erst noch geschaffen werden müssten, ist der konkrete Vorschlag der Studienautoren, dass strukturierte, Mensch- und Maschinen-verständliche Regelungen, die die digitale Umsetzung von Gesetzen unmittelbar legitimieren und rechtssicher gestalten im Rahmen von Gesetzgebungsverfahren anstelle von aus Sicht der IT rein informellen juristischen Regelungen verabschiedet werden sollen. Ziel dieser Studie ist es daher – auch um die Voraussetzungen für das vorgeschlagene Instrument auszuloten und Überzeugungsarbeit für dessen

etwaige Etablierung zu leisten – **Möglichkeiten für die Darstellung von Gesetzen durch strukturierte, Mensch- und Maschinen-verständliche Sprachen** zu analysieren. Es wird also untersucht, ob Gesetze direkt in einer strukturierten Sprache – die Informatik spricht hier von einer „domänenspezifischen Sprache“ – so formuliert werden können, dass sie *für Menschen und Computer geeignet und verständlich sind*. Die Vermeidung eines, aus Sicht der IT, informellen juristischen Textes zu Gunsten einer eindeutigen Regelsprache durch den Gesetzgeber führt nach Auffassung der Studienautoren zu Gesetzesklarheit zwischen allen Beteiligten, erhöht signifikant die Möglichkeit einer digitalen Umsetzung und erleichtert schließlich auch die direkte digitale Umsetzung. Perspektivisch könnten derart präzisierte IT-Regeln auch Unternehmen insbesondere zur Anwendung und für die Implementierung von vor- oder nachgelagerten Schnittstellenlösungen zur Verfügung gestellt werden und so die gesellschaftliche und wirtschaftliche Entwicklung der Digitalisierung unterstützen. Die Studienautoren sind davon überzeugt, dass es langfristig keine Alternative dazu geben wird, als Gesetze zu digitalisieren und mittels einer eindeutigen, strukturierten Sprache in Form von Modellen zu beschreiben. Durch die automatische Transformation dieser Modelle in natürlichsprachlichen Gesetzestext kann mit diesem Ansatz trotzdem sichergestellt werden, dass die Gesetze auch in natürlicher Sprache vorliegen.

Konkret beleuchtet diese Studie existierende wissenschaftliche Ansätze sowie Lösungen aus der Praxis, domänenspezifische Sprachen für die Beschreibung von Gesetzestexten sowie welche Werkzeuge, Systeme und auch ggf. organisatorische und strukturelle Anpassungen im Gesetzgebungsprozess erforderlich sind.

1.3 Aufbau der Kurzstudie

Die Studie ist wie folgt aufgebaut: Kapitel 2 gibt einen Überblick über bestehende wissenschaftliche Ansätze sowie Ansätze aus der Praxis,

² So verfolgt das Gesetz zur Reform des Grundsteuer- und Bewertungsrechts vom 26.11.2019, BGBl. I 2019, 1794 zwar den Gedanken einer automatisierten Veranlagung, was sich insbesondere aus der Gesetzesbegründung im entsprechenden Referentenentwurf entnehmen lässt, ohne jedoch konkret auf die digitale Umsetzung einzugehen.

beschreibt die im weiteren verwendete Terminologie und die wichtigsten Basisprinzipien zum weiteren Verständnis der Studie und diskutiert juristische Vorgaben und Herausforderungen. Kapitel 3 beschreibt die Lösungsansätze für Steuergesetze, diskutiert mögliche Varianten und bietet einen genaueren Einblick in ausge-

wählte Beispiele und deren Realisierung mittels einer domänenspezifischen Sprache. Kapitel 4 beschreibt eine Reihe von Empfehlungen für die Umsetzung. Das letzte Kapitel fasst die Studie zusammen und bietet einen Ausblick auf mögliche nächste Schritte.

2. WISSENSCHAFTLICHER UND PRAKTISCHER HINTERGRUND

2.1 Aktuelle Diskussion in Praxis und Wissenschaft

Dieses Kapitel greift bisherige Diskussionen in der Wissenschaft und Praxis auf, um ggf. Anknüpfungspunkte für einen weiterführenden Austausch zu schaffen.

Betrachtet man **wie Gesetze formuliert sind**, so bestehen hier zunehmend Probleme bezüglich der Digitaltauglichkeit und Automatisierungsfähigkeit des Vollzugs (siehe auch Gutachten im Auftrag des Normenkontrollrats (McKinsey & Company, Inc. 2019)). Hierzu meint bereits Parycek, dass ein „Gesetz so formuliert wird, dass der Text auch für eine Maschine lesbar ist“ (KONSENS magazin, Heft 2, Dez. 2019, Seite 18). Die Möglichkeiten der Automatisierung wird jedoch bedingt durch (bewusst) fehlende Spezifikationen eingeschränkt wie z.B. fehlende Orts- oder Zeitangaben oder Formulierungen, die Ermessensspielräume ermöglichen (Kar et al. 2019). So gehen manche Studienautoren davon aus, dass nur eine Teilautomatisierung möglich ist (Kar et al. 2019, Seite 11). Auch (Barthel 2020) geht davon aus, dass „keine Eins-zu-Eins-Übersetzung der analogen in digitale Prozesse möglich“ ist.

Vorhergehende NEGZ Kurzstudien, z.B. die Studie „Vollzugsorientierte Gesetzgebung durch eine Vollzugssimulationsmaschine“ (Schuppan et al. 2018) gehen auf die „gute“ Formulierung von Gesetzen ein, um die Vollzugsfähigkeit zu ermöglichen. Hierbei wird auch diskutiert, dass beispielsweise das Projekt eGesetzgebung³ den Vollzug von Gesetzen nicht ausreichend im Blick hat. Die Studie schlägt basierend auf (Off, Kühn, Schuppan 2016) drei Schritte vor, mit denen man vom Gesetz zum Vollzug kommt: Der Gesetzestext, die Übersetzung in eine Ontologie und die Transformation in Prozesse mit Hilfe von Referenzprozessmodellen, die dann zur Simulation des Vollzugs verwendet werden. Dieser Ansatz macht aber die Annotation von Gesetzestexten, also das Hinzufügen von zusätzlichen Informationen notwendig, um weiter verarbeitbar zu sein. Dies stellt wiederum einen kritischen Punkt dar, da hierbei Übersetzungsfehler auftreten können.

Bezüglich der **Beschlussfassung von Gesetzen** gibt es bereits eine Vielzahl an Anregungen zur Verbesserung aus der Wissenschaft und Verwaltung. Hierzu erklärt Parycek „Gesetz und Code sollten gemeinsam beschlossen werden,

³ BMI: Projekt eGesetzgebung, URL: https://www.verwaltung-innovativ.de/DE/Gesetzgebung/Projekt_eGesetzgebung/projekt_E_gesetzgebung_node.html, abgerufen am: 1.2.2021.

sodass sich die digitale Welt nicht vom Rechtsstaat löst.“ (KONSENS Magazin, Heft 2, Dez. 2019, Seite 18). Einen Ansatz, wie man die digitale Vollzugstauglichkeit bereits im *Gesetzgebungsprozess* berücksichtigen kann wird in (Kar et al. 2019) beschrieben. Bei diesem Ansatz setzt man auf Entscheidungsregeln, Begriffsdefinitionen und Datenverknüpfungen für die Überführung rechtlicher Vorschriften in ausführbaren Software Code. Im Gegensatz dazu setzt unser Ansatz auf domänenspezifische Sprachen zur Definition dieser Übersetzung in ausführbaren Software Code.

Es gibt aktuell bereits einige Bereiche des Rechts für die IT-Lösungen bestehen, z.B. den Verwaltungsvollzug oder Bereiche des Steuergesetzes, oft auch Legal-Tech genannt.

Das **Föderale Informationsmanagement (FIM)**⁴ verfolgt gemäß eigenen Aussagen das Ziel, leicht verständliche Bürgerinformationen, einheitliche Datenstrukturen für Formulare und standardisierte Prozessvorgaben für den Verwaltungsvollzug bereitzustellen. Die Anwendung der FIM-Methodik soll gemäß IT-Planungsrat und FITKO den Prozess von der Rechtsetzung bis zum Vollzug eines Verwaltungsverfahrens vereinfachen. *Ziel* ist es, den Übersetzungs- und Implementierungsaufwand rechtlicher Vorgaben zu senken. Länder und Kommunen sollen – bezogen auf die redaktionelle und organisatorische Umsetzung eines Verwaltungsverfahrens – nicht mehr für sich alleine agieren müssen. Stattdessen können sie auf Vorarbeiten der nächsthöheren Verwaltungsebene zurückgreifen. Von den drei FIM Bausteinen Leistungen, Datenfelder und Prozesse spielen in dem Kontext der Digitalisierung der Gesetzgebung insbesondere die Datenfelder und Prozesse eine wichtige Rolle. Eine domänenspezifische Sprache ist dabei aber nach dem aktuellen Stand durch FIM nicht vorgesehen. Stattdessen ist angedacht Regeln, soweit überhaupt möglich, als *JavaScript Dateien* zu erfassen. Damit erfüllt das Vorgehen von FIM bereits theoretisch nicht den Anspruch, den Gesetzgebungsprozess gemäß

der Studienintention zu unterstützen. Vielmehr setzt FIM bei der Umsetzung an. Die Wahl von *JavaScript* für die Definition von Regeln erscheint vor dem Hintergrund einer seit über 10 Jahren etablierten DSL im Bereich von ELSTER nicht nachvollziehbar.

(Ruß, Ismer und Margolf, 2019) gehen auf eine Reihe von praktischen Anwendungen ein, u.a. zur Automatisierung im Verfahrensrecht sowie die **Nutzung von Methoden der künstlichen Intelligenz** im Besteuerungsverfahren. Auch wenn die Nutzung von schwacher KI, also der Unterstützung von Menschen bei der Problemlösung, grundsätzlich zu begrüßen ist, so bieten diese Technologien substanzielle Herausforderungen: Eigenheiten in den Trainingsdaten können Ergebnisse ungewollt beeinflussen und die Erklärbarkeit der Ergebnisse ist nicht immer gegeben. Auch aktuelle Parlamentarische Anfragen⁵ beschäftigen sich mit der Frage nach der Maschinenlesbarkeit von Gesetzen. Hierbei wird in der Beantwortung auf **automatisierte Methoden zur Textanalyse** sowie die Nutzung von KI-Methoden eingegangen. Wenngleich solche Methoden als Unterstützung verwendet werden können, so stellt die Interpretation von natürlichsprachlichen Texten nach wie vor eine Aufgabe dar, die nicht ohne händische Nachbesserungen und Überprüfungen auskommt. Zur Verbesserung werden hierfür bspw. Annotationen im Text vorgenommen (Schuppan, Köhl und Off 2019), die jedoch egal ob automatisch, halbautomatisch oder manuell vorgenommen, oft fehlerhaft sind und dadurch die Ergebnisse wieder „verunreinigen“ (Meurers und Müller 2009).

Mit der Digitalisierung von Gesetzen gehen aber auch eine Reihe an **Herausforderungen** einher: Fehlerhafte Interpretationen, fehlende Kenntnisse oder die Anpassung der IT an laufende Gesetzesänderungen.

Die Übersetzung der Gesetzestexte in IT-Systeme, die zuvor bereits durch die Verwaltung interpretiert wurden (siehe Abb. 1), kann leicht zu **fehlerhaften Umsetzungen** in den IT-Systeme-

4 siehe <https://www.bmi.bund.de/DE/themen/moderne-verwaltung/verwaltungsmodernisierung/foerderales-informationsmanagement/foerderales-informationsmanagement-node.html> abgerufen am: 1.2.2021.

5 Siehe Frage 16 in der Antwort der Bundesregierung auf die kleine Anfrage der FDP, BT-Drs. 19/19733, abrufbar unter: <https://dip21.bundestag.de/dip21/btd/19/197/1919733.pdf> abgerufen am: 1.2.2021.

men führen (Ruß, Ismer und Margolf, 2019, S. 413). Auch in (Ruß, Ismer, und Margolf, 2019) werden hierzu einige Meinungen zitiert, die u.a. warnen, dass **Programmierern nicht die Gestaltung des Rechts überlassen werden soll**, sondern vorschlagen, das Gesetz besser an die technischen Gegebenheiten anzupassen. Das widerspricht nicht gänzlich anderen Ansätzen wie z.B. in dem Gutachten des Normenkontrollrats, das davon ausgeht, dass Legisten zu Gesetzesarchitekten und Lösungsdesignern ausgebildet werden sollten (McKinsey & Company, Inc. 2019) was damit eine **Weiterentwicklung des Berufsbilds Juristin/Jurist** bedeutet (Anzinger 2020). Zudem werden interdisziplinäre Teams aus Juristen, Betriebswirten, Informatikern zunehmend wichtig (Ruß, Ismer und Margolf, 2019).

Ein anderer Aspekt ist die grundsätzliche **Digitaltauglichkeit** von Gesetzen. Hierfür werden bspw. Digitaltauglichkeitschecks gefordert (McKinsey & Company, Inc. 2019). Durch die Definition von Modellen, die dann automatisiert in IT-Systeme übertragen werden können, ist bereits ein großer Teil der Digitaltauglichkeit abgedeckt. Die automatisierte Transformation in IT-Systeme bedeutet jedoch nicht, dass Prozesse gänzlich automatisiert werden müssen. Die IT kann auch weiterhin als Assistenz- bzw. Expertensystem zur Unterstützung der Verwaltung eingesetzt werden (Ruß, Ismer und Margolf, 2019).

Gesetze, wie z.B. die Abgabenordnung (AO), müssen auf den gesellschaftlichen Wandel angemessen reagieren können (Bundesministerium der Finanzen (BMF) Monatsbericht November 2019) und entwickeln sich auch in Zukunft laufend weiter (Drüen 2019). Hinzu kommt aber auch, dass die entsprechenden technischen Gegebenheiten geschaffen werden müssen, um diese **Änderungen von Gesetzen in der IT umsetzen** zu können. Aus Erfahrungen der mgm technology partners gmbh Koautoren wurde mit der Nutzung von DSLs diese rasche Reaktionszeit bereits gezeigt: So war es durch die Verwendung von Modellierungstechniken z.B. möglich das Formular zur Steuerlichen Entlastung wegen COVID-19 (Steuererleichterungen aufgrund der Auswirkungen des Coronavirus) innerhalb weniger Tage für bestimmte Bundesländer online zu stellen. Die Handlungsfähigkeit des Staates in Krisensituationen kann also durch die Nutzung von DSLs gleichfalls verbessert werden.

Im Gutachten im Auftrag des Normenkontrollrats (McKinsey & Company, Inc. 2019) werden darüber hinaus **Stärken und Defizite** des aktuellen **Gesetzgebungsprozesses** aufgeführt. Als **Stärken** werden u.a. die umfassende Rechtskonsistenzprüfung, die Berechnung des Erfüllungsaufwands von Gesetzesänderungen, der regelmäßige Austausch mit den Ländern, eine enge Einbindung von organisierten Interessen, eine hohe Sachkompetenz in den Ressorts und ein hoher sprachlicher Standard in den Gesetzen erwähnt. Diese **Stärken können durch die Nutzung von Modellen verstärkt werden**, wenn bspw. Rechtskonsistenzprüfungen auf Modellenebene einfacher durchgeführt werden können. Als **Schwächen** werden 1) die Beschränkung des Lösungsraums durch frühe Festlegung auf konkrete Maßnahmen, 2) der wenig systematische Vergleich von Lösungsalternativen, 3) eine unzureichende Sicht auf den kommunalen Vollzug, 4) die lückenhafte Beteiligung der Betroffenen, 5) fehlende systematische Prüfung der Digitaltauglichkeit, 6) die geringe Nutzung von modernen Kollaborationsformen, 7) der einseitige Fokus auf den Erfüllungsaufwand, 8) die hohe Anforderungsdichte sowie 9) das fehlende umfassende Ausbildungskonzept für Legisten gesehen. Diese Defizite werden in Abschnitt 3.4 im Hinblick auf unseren Ansatz zusammenfassend diskutiert.

2.2 Wichtige IT Begriffe und Prinzipien

Im Folgenden erläutern wir die wichtigsten Begriffe, die für das weitere Verständnis der Studie relevant sind. Es gibt eine Reihe von grundlegenden Ansätzen, um Begrifflichkeiten und deren Beziehungen zu beschreiben und maschinell zu verarbeiten (Blumauer und Pellegrini 2006).

Ansätze wie Glossare, Taxonomien (Sherwin 2009), Thesauri oder Ontologien (Guarino et al. 2009, Gruber 1993, Borst 1997, Smith et al. 2006, Blumauer und Pellegrini 2006) sind (a) nicht ausreichend um bspw. komplexere Zusammenhänge und Abläufe zu beschreiben und (b) ermöglichen zwar die Maschinenlesbarkeit, sind jedoch für menschliche Benutzer ungeeignet. Daher sind für die Beschreibung von Gesetzestexten domänenspezifische Sprachen hilfreich, die es ermöglichen auch auf diese Aspekte Rücksicht zu nehmen.

Einige Herausforderungen im Software Engineering bestehen auf Grund der konzeptionellen Lücke zwischen der Problemstellung (Problem-domäne) und der softwaretechnischen Lösung für dieses Problem (Lösungsdomäne). Die Überwindung dieser Lücke durch handgeschriebenen Code erfordert einen immensen Aufwand und führt zur sogenannten „zufälligen Komplexität“ (France und Rumpe 2007), d.h. Problemen der Lösungsdomäne, die in der Problem-domäne konzeptionell nicht relevant sind. **Modellgetriebene Software Entwicklung** (Völter et al. 2013b) ist ein Überbegriff für Software-Entwicklungsmethodologien, die Modelle als primäre Entwicklungsartefakte einsetzen. Damit ist es möglich die konzeptionelle Lücke und damit die zufällige Komplexität zu verringern. Die Modelle werden dazu verwendet, um beispielsweise Code einer Anwendung zu generieren bzw. diese zu interpretieren und zur Laufzeit eines Systems zu verwenden. Die Automatisierung im Rahmen der modellbasierten Softwareentwicklung (MDSE), wie z.B. die Codegenerierung, erfordert entsprechende **Modellierungssprachen**, die beschreiben, welche Modelle tatsächlich gültig sind, um deren automatisierte und sinnvolle Verarbeitung zu ermöglichen.

Eine **Domänen-spezifische Sprache** oder auch Domänen-spezifische Modellierungssprache genannt ist eine *formale Sprache* (im Gegensatz zur *natürlichen Sprache*), die für ein bestimmtes Problemfeld, die sogenannte Domäne, entworfen und entwickelt wird. Sie ist also auf die Anwendungsdomäne zugeschnitten, um Problemstellungen einer Domäne kompakt zu lösen (Völter et al. 2013a). DSLs zeichnen sich durch weniger Redundanz, eine bessere Lesbarkeit, der Abstraktion von technischem Code und eine leichtere Erlernbarkeit, aufgrund des beschränkten Umfangs aus. Im Gegensatz dazu spricht die Informatik von **universell einsetzbaren Sprachen (eng. General Purpose Language, GPL)**, die für viele Anwendungsgebiete geeignet sind und dadurch meist auch eine hohe Anzahl von Konzepten realisieren. Dazu zählen unter anderem die Standards UML⁶ und SysML⁷.

Die *Verarbeitung natürlicher Sprache* ist und bleibt für IT-Systeme eine Herausforderung, auch wenn Methoden der Computerlinguistik oder künstlichen Intelligenz verwendet werden. So machen z.B. Homonyme und Synonyme von Begriffen oder Referenzen auf andere Satz-teile die automatisierte, fehlerfreie Verarbeitung schwierig und es bedarf menschlichen Eingriffs, um etwaige letzte Fehler auszugleichen. *Formale Sprachen*, wie in DSLs definiert, sind für IT-Systeme gut verarbeitbare Sprachen und ermöglichen die Automatisierung. Die juristische Fachsprache ist durch ihre gute Strukturierung und relativ starke Begriffssicherheit bereits näher an einer formalen Sprache und eignet sich daher gut für die automatisierte Verarbeitung in Form einer DSL.

Für die maschinelle Verarbeitung von Sprachen ist die Syntax und die Semantik der Sprache wichtig. Die **Syntax** einer Sprache (eines Zeichensystems) beschreibt die Regeln, nach denen die Sprachkonstrukte (Zeichen des Zeichensystems) gebildet werden. Mittels einer **Grammatik** definiert man die konkrete Syntax einer Sprache. Verwendet man eine DSL, so verwendet man wohlgeformte Sätze (oder **Modelle** (Stachowiak 1973)), denen zudem eine eindeutige **Semantik** zugeordnet werden kann. Diese Funktionssemantik beschäftigt sich mit den Effekten einer Programmausführung (Thiemann 1994).

2.3 DSLs für Gesetze und Verträge

Die *Unified Legal Language (ULL)* ist eine Beschreibungssprache (Lang 2002), mit der juristische Regelungen modelliert werden können. Die ULL wurde zur Darstellung von steuerrechtlichen Regelungen erprobt. Die methodische Anwendung der ULL besteht darin, sie parallel zu natürlichsprachlichen Gesetzestexten einzusetzen. Zwar ist die Syntax der Gesetzestexte nicht gleich zu ihrer syntaktischen Repräsentation in der ULL, semantisch sollen die ULL Modelle jedoch äquivalent zu den entsprechenden Gesetzestexten sein. Die Abbildung der Geset-

6 Unified Modeling Language, Specification Version 2.5.1, Object Management Group, URL: <https://www.omg.org/spec/UML>, abgerufen am: 17.02.2021.

7 OMG System Modeling Language: Specification Version 1.5, Object Management Group, URL: <https://www.omg.org/spec/SysML/1.5>, abgerufen am 17.2.2021.

zestexte würde dann computergestützte Analysen und Auswertungen erlauben. Die Syntax und Semantik der ULL werden intuitiv beschrieben, ohne die Syntax oder Semantik formal zu definieren oder ein Codegenerierungsschema bzw. eine Referenzimplementierung eines Interpreters einzuführen. Dies erschwert die Implementierung von Analysen und Auswertungen.

Ein Ansatz zur *Modellierung von Verträgen* bezüglich der Ausstrahlungsrechte von verschiedenen Angebotsformen in der Fernsehdomäne wird in (Drave et al. 2020) vorgestellt. Die Modelle bilden korrekte Planungszustände von Ausstrahlungen ab. Die Planungszustände der Ausstrahlungen können bezüglich des Modells automatisch auf Korrektheit überprüft werden. Des Weiteren kann für eine gegebene Ausstrahlung bezüglich des Modells und des aktuellen Planungszustands automatisch eine Menge von Verwertungsfenstern berechnet werden, in denen die gegebene Ausstrahlung verplant werden kann, sodass der entstehende Planungszustand bezüglich des Modells korrekt ist. In (Drave et al. 2020) werden die Syntax und die Semantik informell erklärt, indem die Algorithmen für die Korrektheitsprüfung und Verwertungsfensterberechnung beschrieben werden.

Die *Language for Legal Discourse (LLD)* ist eine Wissensrepräsentationssprache, die speziell für juristische Probleme entwickelt wurde (McCarty 1989). Das praktische Ziel der Sprache besteht in der Anwendung in Kombination mit einem integrierten Analyse- und Planungssystem. Die juristische Problemlösung erfolgt durch Adaption der menschlichen, juristischen Herangehensweise. Die Sprache kann hauptsächlich dazu genutzt werden, um Fakten aus Regeln zu inferieren. Die Syntax der LLD ist künstlich, nicht natürlichsprachlich, und orientiert sich damit nicht an der juristischen Domäne. Eine solche künstliche, domänenfremde Syntax könnte negative Auswirkungen auf die Akzeptanz dieser Sprache bei Domänenexperten haben.

McCarty argumentiert, dass *Natürlichsprachlichkeit* eines der größten Hindernisse ist, um in der Kombination von künstlicher Intelligenz und der Rechtsdomäne Fortschritte zu erzielen (McCarty 2007). McCarty zeigt, dass mithilfe von statistischem Parsen und semantischer Interpretation automatisch bestimmte Informationen aus einer juristischen Argumentation bezüglich eines Sachverhalts berechnet werden können. Die Argumentationen werden somit

nicht modelliert, sondern direkt verarbeitet. In Fällen, in denen klassische Parsing-Algorithmen keine sinnvollen Ergebnisse erzielen, könnte statistisches Parsen eine wichtige Technologie zum Parsen von komplexen juristischen Texten sein.

ROSIE ist eine Programmierumgebung für die Entwicklung von Expertensystemen (Sowizral et al. 1985). ROSIE ist zwar nicht auf die Rechtsdomäne spezialisiert, jedoch kann man mit ROSIE Expertenwissen durch Fakten und Regeln darstellen, deren Syntax sehr stark an natürlicher Sprache angelehnt sind. ROSIE kann zwar an natürlicher Sprache angelehnte Modelle verarbeiten, jedoch bei weitem nicht beliebige natürlichsprachliche Texte. Weiterhin ist ROSIE darauf spezialisiert ein auf den Fakten und Regeln basierendes Expertensystem bereitzustellen und ist somit nicht optimal für alle Anwendungen geeignet. Simple Beispiele für die Anwendung von *ROSIE in der Rechtsdomäne* sind in (Lusti 1986) zu finden.

Ein kürzlich verfasster Aufsatz (Kar et al. 2019) zur digitalen Vollzugstauglichkeit von Gesetzen beschäftigt sich mit Maßnahmen, die zur (Teil-)Automatisierung des Gesetzesvollzugs beitragen. Es wird argumentiert, dass die Digitalisierung von Verwaltungsprozessen Potentiale für einen effizienteren Vollzug birgt. Um diese zu gewährleisten, müssten die Möglichkeiten digitaler Technologien jedoch schon bei der Entwurfsphase von Gesetzen bedacht werden. In (Kar et al. 2019) thematisieren die Autoren, dass die Maschinenverständlichkeit des Rechts in der Entwurfsphase von Gesetzen und Verordnungen beachtet werden sollte. Die vorliegende Studie argumentiert in die gleiche Richtung und schlägt vor, dies unter anderem mit dem Einsatz von DSLs zu erreichen. Im Gegensatz dazu betrachtet (Kar et al. 2019) den Fall, dass juristische Regeln direkt in einer Programmiersprache abgebildet werden. Sowohl die vorliegende Studie als auch (Kar et al. 2019) argumentieren, dass es dafür zwingend notwendig ist, dass ein sinnvoller Umgang mit Mehrdeutigkeiten und Kontextabhängigkeiten von Rechtsbegriffen gefunden wird. Ansonsten können die Rechtsbegriffe nicht als präzise und (teil-)automatisch ausführbare Regeln umgesetzt werden. Es wird argumentiert, dass der Einsatz von DSLs zur Modellierung von Baumdiagrammen und Prozessen sinnvoll wäre. Entscheidungsregeln, welche traditionell in linearen Rechtstexten kodiert sind (Wenn-Dann-Regeln), könnten als Baum-

diagramm dargestellt werden, um die Nachvollziehbarkeit zu erhöhen. Dies würde Softwareentwicklern helfen, da sie klare und präzise Vorgaben bräuchten. Diese Baumdiagramme würden auch als visuelle Verständigungshilfe zwischen unterschiedlichen Expertengruppen dienen können (z.B. Juristen und Softwareentwicklern). Rechtliche Vorschriften könnten explizit durch Prozessabläufe während des Entwurfs modelliert werden. Dadurch ließen sich die Praxistauglichkeit und Aufwände für Anwendungsfälle besser abschätzen. Diese Studie argumentiert für den Einsatz mehrerer spezieller DSLs für verschiedene spezielle Anwendungsfälle.

Die *LegalLanguage* (Soares et. al. 2020) ist eine künstliche Sprache mit deren Benutzung die Struktur von Gesetzestexten per Konstruktion systematisch aufgebaut wird. Durch das Benutzen einer eingeschränkten, künstlichen Sprache würden verschiedene Arten von Beziehungen zwischen Gesetzestextausschnitten besser verwaltet werden können. Die Sprache, wie in (Soares et. al. 2020) vorgestellt, zielt nicht auf die Analyse der Semantik von Gesetzestexten, sondern ausschließlich auf die syntaktische Struktur von Gesetzestexten ab. Das Hinzufügen entsprechender Beschreibungen der Semantiken würde in zukünftigen Arbeiten betrachtet werden (Soares et. al. 2020). Obwohl eine strukturierte und wohldefinierte Beschreibung von Gesetzestexten sinnvoll ist, damit sie auf sinnvoller Weise maschinenverarbeitbar sind, genügt die Vorgabe einer syntaktischen Struktur ohne eine Beschreibung des Verhaltens alleine nicht, um eine sinnvolle (Teil-)Automatisierung der Gesetzesanwendung zu unterstützen.

2.4 Ontologien für die Gesetzesdomäne

Es gibt einige Ansätze zur Konzeptualisierung der Gesetzesdomäne. Die Ansätze versuchen insbesondere die Beziehungen zwischen den Konzepten (eines Teils) der Rechtsdomäne darzustellen, um automatische Berechnungen zur Unterstützung von Rechtsprozessen zu ermöglichen.

Auf einige regelbasierte Expertensysteme für die Gesetzesdomäne wird in (McCarty 1990) verwiesen. Eine anschauliche Einführung zu Expertensystemen in der Rechtsdomäne wird in (Lusti 1986) gegeben. Der Autor folgert, dass eine Automatisierung der Rechtsfindung für Ge-

setze undenkbar wäre, bei denen der Richter nach Ermessen eine Entscheidung fällen kann. Die Validität dieser Folgerung ist fragwürdig, denn auch ein Automatismus, der berechnet, dass ein Ermessen des Richters notwendig ist, wäre sicherlich in vielen Fällen hilfreich. Ein Algorithmus muss nicht immer eine exakte Rechtsfolge bestimmen, sondern könnte auch die Unterspezifikation wie sie im Gesetzestext beschrieben ist, umsetzen, um die Menge aller möglichen Entscheidungen zu berechnen, die im Ermessensspielraum liegen können.

Bench-Capon und Prakken listen einige Herausforderungen und offene Forschungsrichtungen im Kontext der Benutzung von Logik in der Rechtsdomäne auf (Bench-Capon et. al. 2007). Die Autoren argumentieren, dass sich klassische Logik nur begrenzt auf die Rechtsdomäne anwenden lässt und schlagen die Untersuchung der Anwendungsfelder von Beschreibungslogiken in der Rechtsdomäne für zukünftige Arbeiten vor.

Eine Übersicht über verschiedene Ontologien und mögliche Anwendungen in der Rechtsdomäne wird in (Mommers 2010) präsentiert. Der Autor stellt heraus, dass die Rechtsdomäne auf natürlicher Sprache basiert. Obwohl die verwendete Sprache in der Rechtsdomäne von formaler Natur sei, würde sie dennoch nicht so formal wie beispielsweise eine Programmiersprache oder eine Logiksprache sein. Daher gäbe es weder eine formale Syntax noch eine formale Semantik für die verwendete Sprache. Jeder Ansatz, um diese Sprache zu repräsentieren würde an diese Herausforderung stoßen, da für eine Sprache zur Repräsentation mindestens eine formale Syntax gegeben sein müsste, um automatische Analysen zu ermöglichen, und eine formale Semantik gegeben sein müsste, damit die Bedeutungen der Modelle der Sprache eindeutig definiert sind. Die Anwendungsfelder, die der Autor nennt, sind unter anderem (1) Informationsbeschaffung, um Relationen zwischen Objekten zu identifizieren, (2) Rechtsdokumente übersetzen, um Rechtsdokumente von einer Sprache in eine andere zu übersetzen, (3) automatische Klassifizierung und Zusammenfassung, um Informationen aus Dokumenten zu beschaffen, und (4) Entscheidungshilfen für regulierte Rechtsprozesse.

Das *Legal Knowledge Interchange Format* ist eine Kern-Ontologie, die zum Austausch von Wissen zwischen existierenden wissensbasier-

ten Systemen in der Gesetzesdomäne konzipiert wurde (Hoekstra et. al. 2007). Die Hauptnutzen der Ontologie bestünden darin, (1) Daten zwischen existierenden Wissensbasen konvertieren zu können und (2) die Ontologie als einen Wissensrepräsentationsformalismus zu benutzen, der als Teil größerer Architekturen zum Entwickeln von wissensbasierten Systemen in der Gesetzesdomäne verwendet werden könnte.

Ein Vergleich von vier Ontologien für den Entwurf von wissensbasierten Systemen in der Gesetzesdomäne wird in (Visser und Bench-Capon 1998) präsentiert. Die Autoren stellen unter anderem heraus, dass es keine klare Übereinkunft darüber gibt, was die elementaren Konzepte in der Gesetzesdomäne sind. Dies könnte daran liegen, dass verschiedene Arbeiten unterschiedliche Zwecke innerhalb der Gesetzesdomäne verfolgen. Eine weitere Übersicht von Ontologien in der Gesetzesdomäne ist in (Soares et. al. 2020) zu finden.

2.5 Erfahrungen aus der Praxis: ELSTER

Dieser Abschnitt schildert Erfahrungen der mgm technology partners gmbh Koautoren dieser Studie mit der Nutzung von DSLs in der Praxis.

Im Rahmen von *ELSTER (elektronische Steuererklärung)* werden bereits heute von der deutschen Finanzverwaltung detaillierte Fachmodelle des deutschen Steuerrechts erstellt, mit denen die Plausibilisierung der Daten einer Steuererklärung und die Berechnung der Steuermodellbasiert erfolgt. ELSTER hat dazu seit

über 10 Jahren fachliche Modelle der steuerlichen Formulare erfasst, die sowohl als Dokumentation für die verschiedenen Software-Hersteller als auch zur Generierung des Prüf- und Berechnungscodes verwendet werden. Die verwendete domänenspezifische Sprache ist zugeschnitten auf die Plausibilisierung und Berechnung der Steuerdaten und ist sowohl menschenlesbar als auch maschinell verarbeitbar. Die Sprache wird seit über 10 Jahren in ELSTER produktiv eingesetzt und kontinuierlich erweitert (siehe Abbildung 2).

Durch die Verwendung von Fachmodellen, die von der Finanzverwaltung direkt erfasst werden, wird eine digitale Souveränität im Bereich von ELSTER bereits sehr weitgehend unterstützt. Wie bereits zuvor erwähnt, konnte z.B. das Formular zur Steuerlichen Entlastung wegen COVID-19 (Steuererleichterungen aufgrund der Auswirkungen des Coronavirus) innerhalb weniger Tage im MeinElster Portal für bestimmte Bundesländer produktiv geschaltet werden. Die Erfassung des fachlichen Modells zur Steuerentlastung wegen COVID-19 umfasste 86 Felder und 55 fachliche Regeln.

Auch die Plausibilisierung und Berechnung umfangreicher Steuern, wie z.B. für die Körperschaftsteuer (über 2.500 Felder und über 2.800 Regeln), werden von den Steuer-Fachreferaten mit Hilfe der domänenspezifischen Sprache spezifiziert. Aufgrund von regelmäßigen Gesetzesänderungen, wird für die Körperschaftsteuer für jedes Veranlagungsjahr ein eigenes Fachmodell spezifiziert.

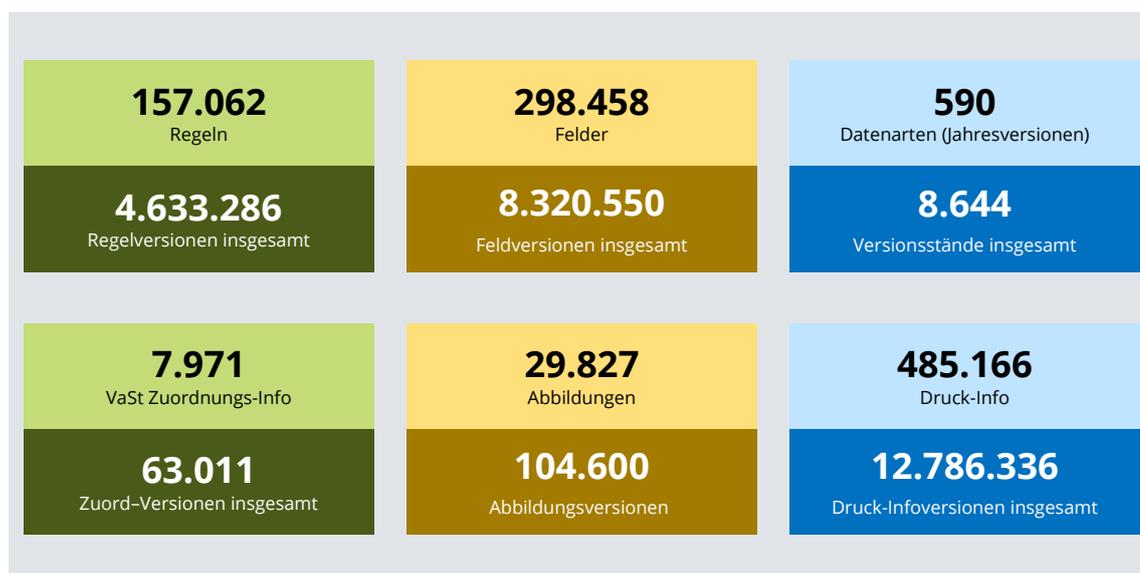


Abbildung 2: Überblick über die Entwicklung von ELSTER im Mai 2020.

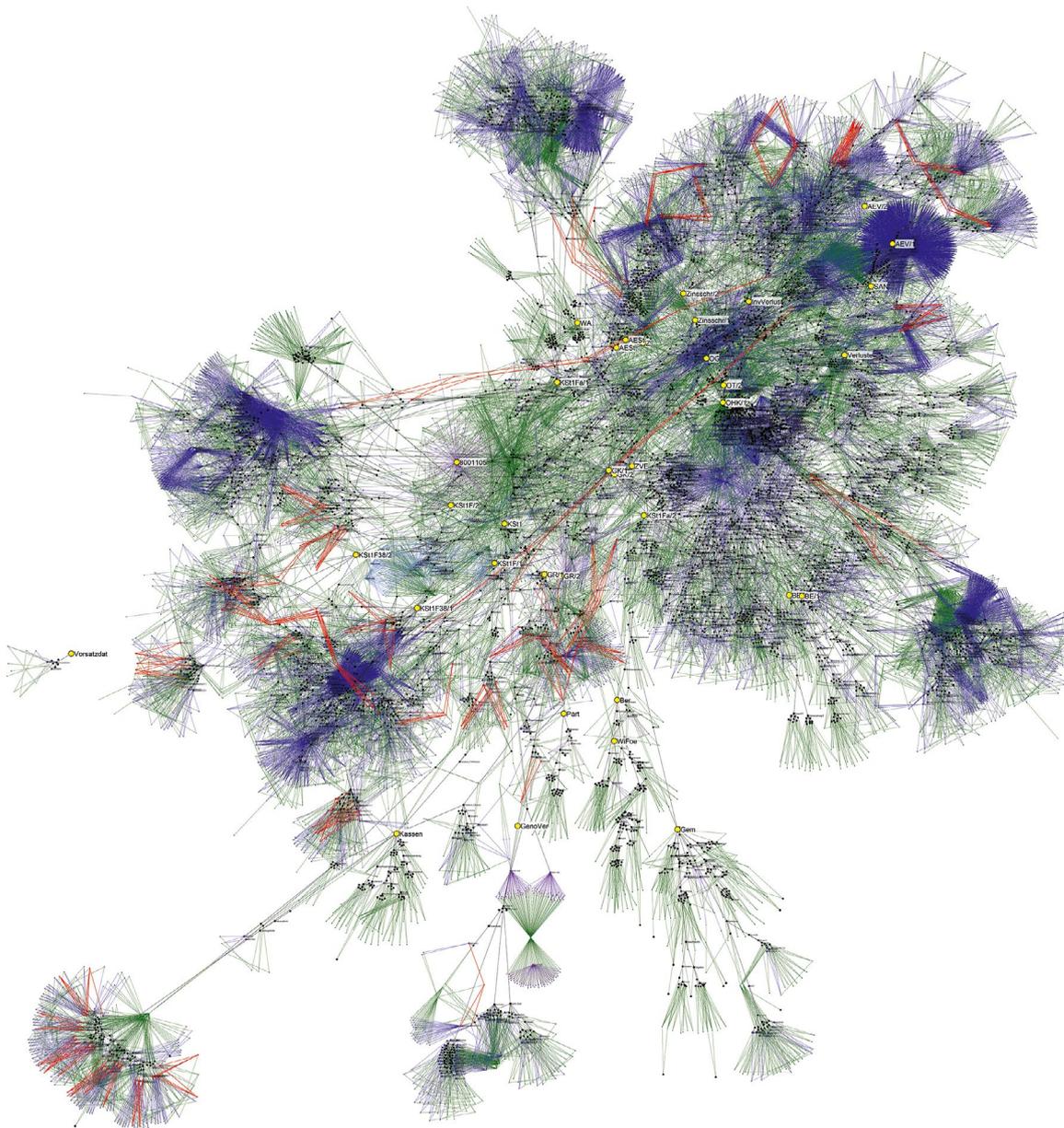


Abbildung 3:
Abhängigkeiten
zwischen Feldern der
Körperschaftsteuer
2018; Punkte bezeich-
nen Felder, Linien
beschreiben fachliche
Zusammenhänge wie
beispielsweise Aus-
schlusskriterien,
Berechnungen oder
Wertvergleiche.

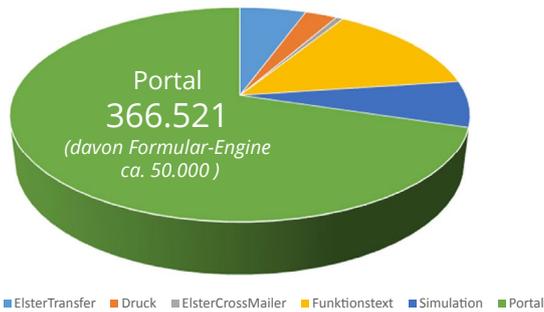
Abbildung 3 zeigt die Abhängigkeiten für die Plausibilitätsprüfungen und Berechnungen zwischen den verschiedenen Feldern einer Körperschaftsteuererklärung. Jeder Punkt entspricht einem Feld in der Steuererklärung, die Linien zwischen den Punkten beschreiben Abhängigkeiten.

Aus den erfassten Fachmodellen wird direkt der Prüf- und Berechnungscode generiert. Im Rahmen von ELSTER wird somit der Weg einer modellbasierten automatisch erstellten Digitalisierungs-Software bereits von der deutschen Steuerverwaltung begangen. Ungeachtet dessen hat gegenwärtig dennoch zwingend eine vorgelagerte, analoge Interpretation der gesetzlichen Regelungen zu erfolgen, die in den elek-

tronischen Formularen in ELSTER umgesetzt werden. Um diesen Arbeitsschritt in den digitalen Prozess eingliedern zu können und auch um Interpretationsfehler zu reduzieren, sollten die von der Verwaltung umzusetzenden Regelungen bereits in menschen- und maschinenverständlicher Sprache durch das Gesetzgebungsverfahren ausgeliefert werden. Durch diese Form der Präzisierung können Abweichungen von der vom Gesetzgeber intendierten Gesetzesanwendung vermieden werden.

Der Umfang der erfassten Modelle und deren Auswirkungen im produktiven Code im MeinElsterPortal lässt sich quantifizieren, indem die Summe der manuell programmierten Zeilen in Bezug zu den aus den Fachmodellen generier-

Mein ELSTER 521.621



ElsterRules-Generale (268) 18.104.205

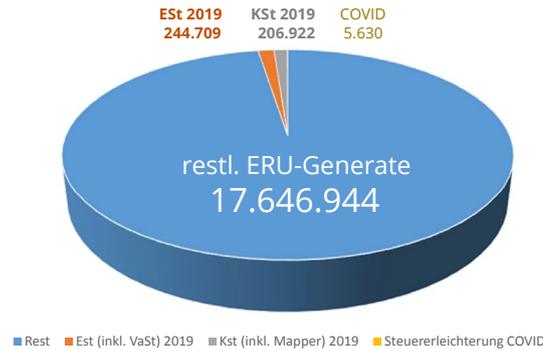


Abbildung 4:
ELSTER Code
Umfang (Lines of
Code) im Mai 2020.

ten Codezeilen gesetzt wird: Abbildung 4 zeigt schematisch, dass im Mai 2020 etwas über eine halbe Millionen Zeilen manuell programmierter Code 18 Millionen Zeilen generiertem Code gegenüberstehen.

2.6 Juristische Rahmenbedingungen für die Digitalisierung der Gesetzgebung und von Gesetzen

2.6.1 Digitalisierungsgrad im Steuerrecht

Wie in Abschnitt 1.1 bereits angesprochen, könnte eine Erklärung für den bereits heute fortgeschrittenen Automatisierungs- und Digitalisierungsgrad in den Prozessen der Finanzverwaltung sicherlich der Umstand sein, dass Steuern grundsätzlich von einer Vielzahl von Steuerpflichtigen im Rahmen von sog. Massenverfahren erhoben werden und insbesondere die schiere Masse an papierbasierten Erklärungen, die regelmäßig in den Finanzämtern eingehen würde, nicht „händisch“ zu stemmen wäre. Zum anderen könnte aber gerade auch die Komplexität des Steuerrechts eine zwingende Automatisierung und Digitalisierung von Prozessen erfordern. So hat bereits der Steuerrechtswissenschaftler Prof. Dr. Klaus Tipke in den 1970er Jahren die These aufgestellt, dass kein Steuerberater und kein Steuerbeamter im Finanzamt auch nur annähernd in der Lage sei, die ganze Stoffmasse der Steuergesetze zu überblicken geschweige denn zu beherrschen

(Tipke 1976). Betrachtet man vor diesem Hintergrund die aktuellen Entwicklungen auf OECD- und EU-Ebene im steuerlichen Bereich, gewinnt man den Eindruck, dass die Komplexität der Steuergesetze bis heute noch weiter zugenommen hat, denn insbesondere mit Fortschreiten der Globalisierung, steigt gleichermaßen der Regelungsbedarf im Bereich des (internationalen) Steuerrechts. Die bereits sehr weit fortgeschrittene Digitalisierung der Steuer ist sicherlich auch auf das Verwaltungsabkommen KONSENS aus dem Jahr 2007 zurückzuführen, das wichtige organisatorische Rahmenbedingungen festgelegt hat. Die Verabschiedung des Gesetzes über die Koordinierung der Entwicklung und des Einsatzes neuer Software der Steuerverwaltung (KONSENS-G) im Jahr 2017 und das Gesetz zur Verbesserung des Onlinezugangs zu Verwaltungsleistungen (OZG), kombiniert mit grundsätzlichen Überlegungen der Effizienz, wird den Automatisierungs- und Digitalisierungsgrad in der Finanzverwaltung sicherlich weiter erhöhen.

Die Anpassung und Einführung von bestimmten Regelungen in der AO, die der Verwaltung und den Steuerpflichtigen neue digitale und automatisierte Möglichkeiten eröffnen, scheint damit ein richtiger Schritt. So kann insbesondere unter gewissen Voraussetzungen die Prüfung von Steuererklärungen⁸ sowie die Steuerfestsetzung⁹ durch die Finanzverwaltung (ausschließlich oder teilweise) automationsgestützt erfolgen. Der Steuerpflichtige hat hingegen die

8 Vgl. § 88 Abs. 5 AO.

9 Vgl. § 155 Abs. 4 AO.

Möglichkeit seine Buchführung und erforderliche Aufzeichnungen auf Datenträgern vorzunehmen¹⁰ oder kann bestimmte Unterlagen ausschließlich auf Datenträgern aufbewahren¹¹.

Allerdings genügt es aus Sicht der Studienautoren nicht, wie bereits in Abschnitt 1.1 und 1.2 angesprochen, dass das Gesetz lediglich die Möglichkeit des Einsatzes von digitalen Prozessen erlaubt. Vielmehr sollte ein weiterer Schritt gegangen werden, mit dem sich die Ausgestaltungen von Gesetzen konkret an digitalen Möglichkeiten ausrichten und entsprechende Regelungen zur digitalen Ausgestaltung enthalten sollten. Als ein aktuelles Beispiel kann die Gesetzesbegründung des Referentenentwurfs zum Grundsteuer-Reformgesetz (GrStRefG)¹², welches als Reaktion des Gesetzgebers auf die BVerfG-Entscheidung¹³ über die teilweise Verfassungswidrigkeit der grundsteuerlichen Bewertung in seiner aktuellen Fassung zu sehen ist, herangezogen werden. In der Gesetzesbegründung wird insbesondere als Ziel des Gesetzes genannt, eine weitgehend einheitliche automationsgestützte Erhebung und eine vereinfachte und ebenfalls automatisierte Bewertung einzuführen sowie elektronisch vorhandene Daten des Immobilienmarkts und der Geodateninfrastruktur zu nutzen.¹⁴ Trotz der genannten Zielsetzung fehlt es im Gesetz selbst – soweit ersichtlich – an konkreten Regelungen zur digitalen Umsetzung. So merkt der bekannte Verfassungs- und Steuerrechtler Gregor Kirchhof dazu an, dass ungeachtet einer digitalen Umsetzbarkeit des grundsteuerlichen Bewertungsverfahrens, Fragen zur Digitalisierung nach den beizubringenden Daten nicht beantwortet seien.¹⁵ Eine Konkretisierung der digitalen Umsetzung des Bundesmodells hätte ggf. dazu geführt, dass ein einfacheres Bewertungsverfahren im Bundesmodell etabliert worden wäre und möglicherweise weniger Länder An-

strengungen anstellen würden, abweichende Regelungen zum Bundesmodell auf Grundlage von Art. 72 Abs. 3 Nr. 7 GG zu treffen. Eine Ausarbeitung eines präzisen Modells unter Einsatz einer präzise definierten DSL im Rahmen des Gesetzgebungsverfahrens, wie dies in Abschnitt 2.6.2.6 dargestellt ist, hätte ggf. ebenfalls Einfluss auf das Ergebnis der Reform der Grundsteuer gehabt.

Es ist allerdings anzumerken, dass insbesondere beim Bewertungsgesetz, welches durch das GrStRefG angepasst worden ist, ein solches Modell (auch nachträglich) einfach einzuführen wäre, da das Gesetz zumindest für die Bewertung von Grundstücken bereits heute schon genaue Anweisungen für die Berechnung der Grundsteuerwerte gibt. Vor Erstellung des Modells sollte allerdings Einigung über eine zugehörige, präzise definierte, allgemein akzeptierte Modellierungssprache erfolgen.

2.6.2 Das Gesetzgebungsverfahren

2.6.2.1. Ausarbeitung von Modellen im Gesetzgebungsverfahren

Neben der Untersuchung nach konkreten technischen Anforderungen an Modelle ist die Erläuterung der Möglichkeiten der (initialen) Ausarbeitung von Modellen im Rahmen des Gesetzgebungsverfahrens nicht minder wichtig. Schließlich ist – zumindest für einen gewissen Übergangszeitraum – davon auszugehen, dass als Vorarbeit für Modelle weiterhin ein aus Sicht der IT informeller juristischer Gesetzestext als Vorlage ausgearbeitet werden wird. Je früher im Rahmen des Gesetzgebungsverfahrens die jeweiligen Modelle erarbeitet werden, desto weniger Übereinstimmungsprobleme sollten zwischen dem informell juristischen Text und den Modellen zu erwarten sein. Auch wenn eine

10 Vgl. § 146 Abs. 5 AO.

11 Vgl. § 147 Abs. 2 AO.

12 Gesetz zur Reform des Grundsteuer- und Bewertungsrechts vom 26.11.2019, BGBl. I 2019, 1794.

13 BVerfG vom 10.4.2018, 1 BvL 11/14, 1 BvR 889/12, 1 BvR 639/11, 1 BvL 1/15, 1 BvL 12/14, BVerfGE 148, 147.

14 Vgl. Referentenentwurf des Gesetzes zur Reform des Grundsteuer- und Bewertungsrechts, S. 86, 89, 90, 91, 92.

15 Vgl. Kirchhof, Die Reform der Grundsteuer und das Maß des Grundgesetzes, Vorläufige Ergebnisse eines Gutachtens im Auftrag des Zentralen Immobilien Ausschusses e. V., III. Das „wertabhängige Mietmodell“ verletzt das Grundgesetz, 7. Verfassungsverstoß im Vollzug, Verwaltungslasten, Digitalisierung, URL: https://zia-deutschland.de/wp-content/uploads/2021/05/G_Kirchhof_Gutachten_ZIA_10.1.20191.pdf, abgerufen am: 1.2.2021.

entsprechende Anwendung von Modellen auf Landesebene und in Kommunen nicht ausgeschlossen wird, wird im Folgenden der Fokus auf die Gesetzgebung auf Bundesebene gelegt und in den Abschnitten 2.6.2.2 bis 2.6.2.5 zunächst der entsprechende Prozess des Gesetzgebungsverfahrens auf Basis der einschlägigen Regelungen im GG, der GOBT, GOBR, GOBReg sowie GGO wiedergegeben, um in Abschnitt 2.6.2.6 Handlungsempfehlungen für die Erstellung der DSL-Vorlage im Rahmen des Gesetzgebungsverfahrens ableiten zu können.

2.6.2.2. Gesetzesvorlage

Eine Gesetzesvorlage¹⁶ kann durch die Bundesregierung, den Bundesrat oder aus der Mitte des Bundestages initiiert werden.¹⁷ Die meisten Gesetzesvorlagen werden durch die Bundesregierung durch Einbringung beim Deutschen Bundestag initiiert (Gröpl 2017). Dies ist überwiegend bei Steuergesetzesvorlagen der Fall, bei denen dem Bund die sog. konkurrierende Gesetzgebung zusteht und diese Gesetzesvorlagen der Zustimmung des Bundesrats bedürfen, wenn von der Gesetzesvorlage Steuern betroffen sind, deren Aufkommen den Ländern oder den Gemeinden ganz oder zum Teil zufließen.¹⁸ Aus Sicht des Bundesrats handelt es sich um sog. Zustimmungsgesetze (s.u.). Im Folgenden beschränken sich die Ausführungen auf Zustimmungsgesetze, bei denen dem Bund die konkurrierende Gesetzgebung zusteht.

In der Praxis erarbeitet die Bundesregierung einen Gesetzesentwurf nicht selbst, vielmehr wird dieser im Regelfall durch die Bundesministerien und dort insbesondere auf Referatsebene erarbeitet.¹⁹ Ein solcher Entwurf beinhaltet neue gesetzliche Regelungen und/oder Änderungen bestehender gesetzlicher Regelungen mit einer entsprechenden Begründung (Gesetzesbegründung)²⁰. Nach Fertigstellung des Referentenentwurfs erfolgt dessen Weiterleitung – mit der Einladung eine fristgerechte Stellungnahme abzugeben – an Zentral- und Gesamtverbände sowie an Fachkreise, die auf Bundesebene bestehen, wenn deren Belange berührt sind.²¹ Ggf. kann eine mündliche Anhörung erfolgen.²²

In einem nächsten Schritt wird der Referentenentwurf der Bundesregierung zum Beschluss vorgelegt, in dessen Anschreiben zur Kabinettsvorlage²³ abweichende Meinungen der oben genannten Zentral- und Gesamtverbände sowie Fachkreise, die auf Bundesebene bestehen, anzugeben sind.²⁴ Dadurch wird dem federführenden Ressort der Bundesregierung²⁵ die Gelegenheit gegeben, die Interessen der Betroffenen zu berücksichtigen sowie mögliche Fehler im Gesetzentwurf oder unzutreffende Sachverhaltsannahmen möglichst frühzeitig zu korrigieren.²⁶ Über den Referentenentwurf wird ein sog. Kabinettsbeschluss gefasst.²⁷ Bei Zustandekommen eines Kabinettsbeschlusses über die Einbringung des Gesetzesentwurfs wird das Gesetzesvorhaben zu einem förmlichen Regie-

16 Gesetzesvorlagen werden als Drucksache veröffentlicht.

17 Vgl. Art. 76 Abs. 1 GG.

18 Vgl. Art. 105 Abs. 2, 3 GG.

19 Vgl. BMF, URL: https://www.bundesfinanzministerium.de/Web/DE/Service/Gesetze_Gesetzesvorhaben/Gesetze_Gesetzgebungsvorhaben.html, abgerufen am: 4.1.2021.

20 Gem. § 76 Absatz 2 GOBT muss ein Gesetzesentwurf eine Begründung enthalten.

21 Vgl. § 47 Abs. 1, 3 GGO.

22 Vgl. § 47 Abs. 5 GGO.

23 Die Bundesregierung, welche auch als (Bundes-)Kabinett bezeichnet wird, besteht aus den Bundesministerinnen und -ministern und der Bundeskanzlerin bzw. dem Bundeskanzler.

24 Vgl. § 51 Nr. 4 GGO.

25 Bei der Steuergesetzgebung ist dies oftmals die Finanzministerin/der Finanzminister und ggf. die Wirtschaftsministerin/der Wirtschaftsminister.

26 Vgl. Wissenschaftlicher Dienst des Deutschen Bundestags, Beteiligung von Verbänden an der Vorbereitung von Gesetzesvorlagen der Bundesregierung, WD 3 – 3000 – 030/15, URL: <https://www.bundestag.de/resource/blob/405282/1eea81a3a2736258e123a8359ac84fa3/wd-3-030-15-pdf-data.pdf>, abgerufen am: 1.2.2021 mit Verweis auf Maaßen, Gesetzesinitiativen der Bundesregierung, in: Kluth/Krings (Hrsg.), Gesetzgebung (2014), § 8 Rn. 66.

27 Vgl. § 15 Abs. 1 BSt. a) GOBReg i.V.m. § 24 Abs. 2 Satz 1 GOBReg.

rungsentwurf, sodass von nun an nicht mehr vom Referentenentwurf, sondern vom Regierungsentwurf eines Gesetzes gesprochen wird. Der Regierungsentwurf wird sodann dem Bundesrat zugeleitet, um diesem innerhalb von sechs Wochen die Möglichkeit der Stellungnahme zum Regierungsentwurf zu geben.²⁸

Ein Gesetzesentwurf kann alternativ auch durch Mehrheitsbeschluss durch den Bundesrat erfolgen.²⁹ Der Bundesrat hat diesen innerhalb von sechs Wochen durch die Bundesregierung an den Bundestag zuzuleiten.³⁰ Die Bundesregierung soll zum Entwurf Stellung nehmen.³¹

Das Grundgesetz enthält keine speziellen Regelungen zur Einbringung eines Gesetzesentwurfs durch die Mitte des Bundestages.

2.6.2.3. Beratung und Gesetzesbeschluss im Bundestag

Gesetze werden im Bundestag beschlossen³² und zwar grundsätzlich nachdem drei Beratungen (Lesungen) über den Gesetzesentwurf vorangegangen sind.³³

Nach der ersten Beratung wird der Gesetzesentwurf zur genaueren Prüfung und Bearbeitung an den zuständigen Ausschuss weitergeleitet.³⁴ Bei steuerlichen Gesetzesvorhaben ist dabei grundsätzlich der Finanzausschuss (7. Ausschuss) zuständig, der sich aus Bundestagspolitikern mit Expertise im Bereich Finan-

zen zusammensetzt. Im Rahmen der Detailberatung im Ausschuss können (externe) Experten im Rahmen von meist öffentlichen Anhörungen eingeladen und angehört werden. Der Ausschuss schließt seine Arbeit mit einem schriftlichen Bericht und einer Beschlussempfehlung³⁵ zur Abstimmung im Bundestag ab, welche dem Plenum zur zweiten Beratung vorgelegt werden.

In der zweiten Beratung kommt es zur Aussprache über den Entwurf, den Bericht des Ausschusses und dessen Änderungsvorschläge.³⁶ Dann wird über jede Einzelbestimmung abgestimmt.³⁷ Jeder Abgeordnete kann grundsätzlich Änderungsanträge stellen.³⁸ Darauf folgt die dritte Beratung, in der unter gewissen Voraussetzungen ausschließlich Änderungsanträge zu denjenigen Bestimmungen gestellt werden können, zu denen in zweiter Beratung Änderungen beschlossen wurden.³⁹ Grundsätzlich wird nach der dritten Beratung über den Gesetzentwurf abgestimmt (Schlussabstimmung).⁴⁰

2.6.2.4. Beratung und Zustimmungentscheidung im Bundesrat

Nachdem das Gesetz im Bundestag beschlossen wurde, leitet der Bundestagspräsident den sog. Gesetzesbeschluss an den Bundesrat weiter.⁴¹ Analog wie im Bundestag gibt es auch im Bundesrat Fachausschüsse, welche die Beschlussfassung des Bundesrats vorbereiten.⁴² Bei steuerlichen Gesetzesvorhaben ist dabei

28 Vgl. Art. 76 Absatz 2 Satz 1, 2 GG.

29 Vgl. Art. 52 Abs. 3 Satz 1 GG.

30 Vgl. Art. 76 Abs. 3 Satz 1 GG.

31 Vgl. Art. 76 Abs. 3 Satz 2 GG.

32 Vgl. Art. 77 Abs. 1 Satz 1 GG.

33 Vgl. § 78 Abs. 1 Satz 1 GOBT.

34 Vgl. § 80 Abs. 1 Satz 1 GOBT.

35 Bericht und Beschlussempfehlung werden als Bundestags-Drucksache veröffentlicht.

36 Vgl. § 81 Abs. 1 Satz 1, Abs. 2 Satz 1 GOBT.

37 Vgl. § 81 Abs. 2 Satz 2 GOBT.

38 Vgl. § 82 Abs. 1, 2 GOBT.

39 Vgl. § 85 Abs. 1 GOBT.

40 Vgl. § 86 GOBT.

41 Vgl. Art. 77 Abs. 1 Satz 2 GG.

42 Vgl. § 39 Abs. 1 GOBR.

grundsätzlich der Finanzausschuss zuständig.⁴³ Über die Beschlussempfehlung des Ausschusses stimmt das Plenum des Bundesrats ab.⁴⁴

Bei Zustimmungsgesetzen ist nach dem Grundgesetz die Zustimmung des Bundesrats nötig.⁴⁵ Stimmt der Bundesrat nicht zu, kann das Gesetz ggf. im Rahmen eines Vermittlungsverfahrens und erneuter Beratungen verabschiedet werden.

Bei Zustimmung des Bundesrats kommt das vom Bundestag beschlossene Gesetz zustande.⁴⁶

2.6.2.5. Abschlussverfahren

Nach der Verabschiedung eines Gesetzes durch Bundestag und Bundesrat wird es üblicherweise dem zuständigen Bundesministerium bzw. den zuständigen Bundesministerien und im Anschluss der Bundeskanzlerin/dem Bundeskanzler zugeleitet, sodass von beiden Seiten eine Gegenzeichnung des Gesetzes erfolgt.⁴⁷

Abschließend erhält der Bundespräsident das Gesetz, um zu prüfen, ob es verfassungskonform ist bzw. verfassungskonform zustande gekommen ist sowie zur Unterzeichnung und Ausfertigung. Wenn keine Bedenken bestehen, unterzeichnet der Bundespräsident das Gesetz, sodass abschließend das ausgefertigte Gesetz im Bundesgesetzblatt verkündet wird.⁴⁸ Die Verkündung im Bundesgesetzblatt wird in der Praxis durch das Bundesministerium der Justiz und für Verbraucherschutz (BMJV) übernommen. Damit kann es am im Gesetz festgelegten Stichtag bzw. bei Fehlen einer Regelung zum in Kraft treten, am 14. Tag nach der Ausgabe des Bundesgesetzblatts in Kraft treten.⁴⁹

2.6.2.6. Handlungsempfehlungen für die Erstellung von Modellen im Rahmen des Gesetzgebungsverfahrens

Der erläuterte Prozess des Gesetzgebungsverfahrens bietet an verschiedenen Stellen die Möglichkeit entsprechende Modelle zu erarbeiten. Es sollte dabei grundsätzlich bedacht werden, dass insbesondere die Politikerinnen und Politiker im Bundestag und Bundesrat in der Mehrheit keinen IT-technischen Hintergrund haben. Um präzise Modelle mit eindeutigen Bedeutungen zu erhalten, muss in einem vorangehenden Schritt durch interdisziplinäre Expertenteams die entsprechende DSL zur Formulierung der Modelle der Gesetze erarbeitet und umgesetzt werden.

Bei Gesetzesinitiativen der Bundesregierung sollte es daher weiterhin möglich sein, dass die Regierungsmitglieder den Anstoß für die Erstellung eines Referentenentwurfs im Rahmen eines informellen Textes ausarbeiten und sich zunächst nicht zwingend einer DSL bedienen müssen. Da der entsprechende Referentenentwurf im zuständigen Ministerium ausgearbeitet wird, scheint es sinnvoll, dass auf Referatsebene Juristen zusammen mit IT-Experten die Erstellung des Referentenentwurf-Modells ausarbeiten. Bei Erstellung der Modelle in diesem frühen Stadium kann gewährleistet werden, dass über Inhalt und digitale Umsetzung des Gesetzes in einem Schritt nachgedacht wird und es im Rahmen des Gesetzgebungsverfahrens zu keiner Abweichung der Modelle von einem vorgelagert erstellten reinen juristischen Gesetzestext kommen kann. Zudem können an dieser Stelle des Gesetzgebungsverfahrens bereits auch Auffassungen von Zentral- und Gesamtverbänden sowie von Fachkreisen, die sich mit Fragen der Digitalisierung beschäftigen, konkret zum Referentenentwurf-Modell eingeholt werden. In der Gesetzesbegründung des Referentenentwurfs

43 Vgl. Art. 52 Abs. 4 GG.

44 Vgl. Art. 77 Abs. 2a GG.

45 Vgl. Art. 105 Abs. 3 GG.

46 Vgl. Art. 78 GG.

47 Vgl. Art. 58 Satz 1 GG.

48 Vgl. Art. 82 Abs. 1 Satz 1 GG.

49 Vgl. Art. 82 Abs. 2 GG.

könnte die Ausgestaltung der Modelle in informeller Textform erfolgen.

Bei Gesetzesinitiativen durch den Bundesrat oder durch die Mitte des Bundestags besteht hingegen nicht die Möglichkeit Bundesministerien in den Entwurf der Modelle einzubinden. Insbesondere aus diesem Grund sollte daher in den zuständigen Ausschüssen (für steuerliche Fragen üblicherweise der Finanzausschuss) in Bundestag und Bundesrat, welche regelmäßig auch externe Experten anhören, Expertise im Bereich IT aufgebaut werden, um ein Referententwurf-Modell gegenprüfen zu können.

Am Prozess des Gesetzesbeschlusses im Bundestag und der Zustimmungentscheidung im Bundesrat dürfte sich durch Einsatz von Modellen anstatt eines rein juristischen Gesetzestextes wenig ändern, da Gesetzestexte ungeachtet einer juristischen Form oder DSL i.d.R. ohnehin recht komplex sind und die einzelne Politikerin bzw. der einzelne Politiker auf die Auffassung der Ausschüsse in Bundestag- und Bundesrat und Unterstützung ihrer bzw. seiner wissenschaftlichen Mitarbeiter, Referenten und Büroleiter angewiesen ist. So erscheint es sicher sinnvoll, dass zusätzlich eine gewisse Modellierungs-Expertise im Mitarbeiterkreis eines jeden Abgeordneten aufgebaut werden sollte.

2.6.3 Die juristische Gesetzesauslegung als Basis für eine DSL

Nachdem in Abschnitt 2.6.2.6 der (initiale) Einsatzort von Modellen erläutert wurde, wird in Abschnitt 2.6.3 dargelegt, dass die Modelle auch Ergebnis einer juristischen Gesetzesauslegung sein müssen, um den Zweck eines leicht digital umzusetzenden Gesetzes erfüllen zu können.

Im Gegensatz zu im anglo-amerikanischen Raum präsenten Rechtsordnungen des **Fallrechts (eng. Case Law)**, die richterliche Entscheidungen zu konkreten Rechtsfällen beinhalten, daher sehr ausführlich gehalten sind, einen hohen Umfang besitzen, was teilweise zu Wiederholungen im Gesetz führt und eher anfällig für Inkonsistenzen im Gesetz ist, ist im europäischen Raum, so auch in Deutschland, die Rechtsordnung durch **Kodex-Gesetze (eng. Code Law)** üblich. Das Code Law zeichnet sich zumindest in der Theorie durch mehr oder weniger abstrakt gehaltene kurze Formulierungen,

Generalregelungen und eine hohe Übersichtlichkeit aus, was allerdings einer stärkeren Auslegung bedarf und die Problematik von ungeregelten Sachverhalten mit sich bringt.

Die Struktur des Code Laws ist sicherlich näher an einer IT-Sprache als die des Fallrechts (siehe auch Abschnitt 2.2 und 2.3). Um zu gewährleisten, dass ein Modell die rechtssichere digitale Umsetzung des Gesetzeswillens beinhaltet, müssen bei der Erstellung eines Modells, das auf Basis eines Entwurfs in Form eines juristischen Textes entstehen soll, sämtliche Methoden der juristischen Gesetzesauslegung berücksichtigt werden. Dazu zählt die Auslegung nach dem Wortlaut (grammatische Auslegung), nach der Systematik im Gesetz (Kontext zu anderen Normen), nach der Gesetzeshistorie (Beachtung von ähnlichen Vorgängerregelungen) und nach dem Sinn und Zweck des Gesetzes (teleologische Auslegung). In der juristischen Praxis kann die Anwendung dieser verschiedenen Methoden jedoch insbesondere bei komplexen Regelungen zu abweichenden bis sich gar widersprechenden Ergebnissen und damit zu Unsicherheiten führen. Zur Annäherung von juristischen (Entwurfs)Texten und Modellen siehe Abschnitt 3.1.2.

In Abgrenzung zu diesem Vorschlag scheint aufgrund der Auslegungsproblematik eines juristischen Textes die rechtssichere digitale Umsetzung eines Gesetzes durch die Verwaltung, wie es heute der Fall ist, fast unmöglich. Dies liegt oftmals auch daran, dass Gesetze aus politischen Gründen beabsichtigt unpräzise formuliert sind. Das Plädoyer für die Forderung nach Vorgabe von konkreten, verbindlichen Regelungen zur digitalen Gesetzesumsetzung mittels DSLs bereits im Rahmen der Gesetzgebungsverfahren, sodass die Verwaltung in höchstmöglichem Maße rechtssicher agieren und Auslegungsfragen begegnen kann, wird durch dieses Argument nochmals verstärkt.

Auch wenn die Umsetzung einer solchen Forderung sicherlich außergerichtliche und vor (Finanz-)Gerichten geführten Rechtsbehelfsverfahren zur Gesetzesauslegung reduzieren würde, so verhindert die Nutzung von DSLs nicht, dass an sich verfassungswidrige Gesetznormen in Modellen definiert werden könnten.

2.6.4 Vom Kodex-Gesetz zur DSL und Modellen

Wie bereits in Abschnitt 2.6.3 angedeutet, ist der Aufbau eines informellen, juristischen Textes des Code Laws und der eines Programmcodes oftmals relativ ähnlich.

So finden sich zu Beginn eines Gesetzes oftmals bloße Definitionen, die dazu dienen, Wiederholungen von häufig gebrauchten Erklärungen mittels Verweis auf diese vorstehenden Definitionen zu vermeiden und den Gesetzesumfang somit zu verringern. Definitionsbestimmungen können auch gesetzesübergreifend gelten, so definiert § 3 Abs. 1 AO den Begriff „Steuern“, sodass dieser nicht in jedem Einzelsteuergesetz einzeln erläutert werden muss. Die AO selbst bedient sich wiederum für die Berechnung von Fristen und für die Bestimmung von Terminen ausgewählten Regelungen des Bürgerlichen Gesetzbuchs (BGB), indem die §§ 187 bis 193 BGB entsprechend anzuwenden sind, soweit keine abweichende Regelung in der AO einschlägig ist.⁵⁰ So werden die Regelungen zu steuerlichen Fristen und Terminen in der AO möglichst knapp gehalten und müssen nicht in jedem Einzelsteuergesetz aufgeführt werden. Die AO bildet somit eine Teilkodifikation, d.h. ein Mantelgesetz des allgemeinen Steuerrechts, in der im ersten Teil insbesondere Grundbegriffe des Steuerrechts, im zweiten Teil das Steuerschuldrecht, in §§ 16ff. AO und im dritten bis siebten Teil das Steuerverfahren und abschließend im achten Teil das Steuerstraf- und Steuerordnungswidrigkeitenrecht geregelt sind.⁵¹ Auch wenn die Bezeichnung als Steuergrundgesetz juristisch nicht korrekt erscheint, da sich die Grundnormen der Steuerrechtsordnung aus dem Grundgesetz ergeben⁵², gelten die Regelungen der AO für eine Vielzahl von Steuergesetzen⁵³.

Die genannten Regelungen der AO gelten beispielsweise für das Besteuerungsverfahren für

die Körperschaftsteuer.⁵⁴ Durch „Vorziehen“ von allgemeinen Regelungen in die AO und Verweis auf Regelungen des Einkommensteuergesetzes⁵⁵, kann der Umfang des Körperschaftsteuergesetz (KStG) vergleichsweise gering gehalten werden, sodass sich der Inhalt des Gesetzes selbst im Wesentlichen auf die Steuerpflicht, ergänzende Regelungen zur Einkommensermittlung und Spezialvorschriften beschränken kann. Ungeachtet des Umfangs handelt es sich dennoch um ein äußerst komplexes Gesetz.

Die häufigsten juristischen Normen erläutern jedoch einen (abstrakten) Tatbestand und eine damit verbundene Rechtsfolge, welche in einer Wenn-Dann-Beziehung, die man auch aus der Programmiersprache kennt, zueinanderstehen. Eine solches Schema enthält beispielsweise § 8 Abs. 2 KStG: Sofern der Tatbestand der unbeschränkten Steuerpflicht i.S.d. § 1 Abs. 1 Nr. 1-3 KStG erfüllt ist, soll die Rechtsfolge eintreten bzw. gelten, dass alle Einkünfte als Einkünfte aus Gewerbebetrieb zu behandeln sind. Dabei trifft § 8 Abs. 2 KStG selbst keine Aussage darüber, unter welchen Tatbestandsvoraussetzungen die unbeschränkte Steuerpflicht vorliegt, sondern verweist insoweit wörtlich auf § 1 Abs. 1 Nr. 1-3 KStG, sodass bei Vorliegen der dort aufgeführten Körperschaften, Personenvereinigungen und Vermögensmassen mit Geschäftsleitung oder Sitz im Inland (Tatbestand) die unbeschränkte Steuerpflicht (Rechtsfolge) eintritt.

Eine solche Verknüpfung ist in deutschen (Steuer-)Gesetzen häufig zu finden, sodass mit Fortschreiten der Paragraphennummerierung eine immer komplexere Aneinanderreihung von Wenn-Dann-Beziehungen vorliegen kann.

In der juristischen und steuerlichen Praxis ist eine wichtige Aufgabe einen konkret vorliegenden Lebenssachverhalt zu würdigen und ihn einem im Gesetz aufgeführten abstrakten Tatbestand zuzuordnen und die entsprechenden

50 Vgl. § 108 Abs. 1 AO.

51 Vgl. Lang in Tipke/Lang, Steuerrecht, 20. Auflage, § 2, Rn. 10.

52 Vgl. Lang in Tipke/Lang, Steuerrecht, 20. Auflage, § 2, Rn. 12.

53 Vgl. Lang in Tipke/Lang, Steuerrecht, 20. Auflage, § 2, Rn. 13.

54 Vgl. § 1 Abs. 1 Satz 1 AO.

55 Vgl. insbesondere § 8 Abs. 1 Satz 1 KStG und § 26 Abs. 1 KStG.

Rechtsfolgen zu bestimmen bzw. darzulegen, ob ein Tatbestand gerade nicht erfüllt ist, mit der Folge, dass eine gewisse Rechtsfolge eben nicht eintritt. Modelle müssen diese Eigenschaft aus juristischer Sicht ebenfalls erfüllen können.

2.6.5 Verhältnis von juristischen Gesetzestexten zu DSLs bzw. Modellen

Unter der Bedingung, dass der juristische, informelle Gesetzestext und die Modelle (welche durch die DSLs definiert sind) sowie die Auslegung des Gesetzestextes und die Interpretation der Modelle zueinander konsistent sind, wird eine juristische Auslegung des Gesetzestextes oder der Modelle effektiv zum selben Ergebnis führen. Soweit der Gesetzestext aus den Modellen generiert wird und eine entsprechende Konsistenz sichergestellt ist, sollten daher Überlegungen nicht von vornherein ausgeschlossen werden, die eine Verabschiedung von Modellen und nicht mehr von juristischen Gesetzestexten im Rahmen des Gesetzgebungsverfahrens vorsehen. Hierbei sind jedoch insbesondere vorab verfassungsrechtliche Fragen zu beantworten. So ist zu klären, ob eine Verabschiedung von

Gesetzen in nicht natürlicher deutscher Sprache grundsätzlich möglich erscheint.⁵⁶ Diese Frage vermögen wir im Rahmen dieser Studie nicht zu beantworten.

Ggf. könnten für einen Übergangszeitraum (auch aus Akzeptanzgründen) die aus den Modellen generierten Gesetzestexte (und nicht die Modelle selbst) verabschiedet werden. Selbst bei Vorliegen der oben beschriebenen Konsistenz zwischen Gesetzestext und Modellen und bei verfassungsrechtlicher Unbedenklichkeit bei der Verabschiedung von Modellen, sollten ergänzend folgende Voraussetzungen für die Verabschiedung von Modellen erfüllt sein:

- (a) Die Nutzung von DSLs wird von allen relevanten Nutzergruppen akzeptiert.
- (b) Es haben sich bezüglich der Nutzung der DSLs Expertisen in allen relevanten Nutzergruppen herausgebildet.
- (c) Es hat sich herausgestellt, dass die Modelle und deren Auswertungen ihren Zweck erfüllen und damit die betroffenen Prozesse verbessern.

3. ERGEBNISSE: LÖSUNGSANSÄTZE FÜR STEUERGESetze UND DISKUSSION MÖGLICHER VARIANTEN

Im Folgenden diskutieren wir mögliche Lösungsvarianten für die Darstellung von Gesetzen durch strukturierte, Mensch- und Maschinen-verständliche Sprachen sowie eine DSL zur

Abbildung von gesetzlichen Steuerberechnungen und Validierung, eine DSL zur Modellierung von gesetzlichen Fristen und das Zusammenspiel der unterschiedlichen DSLs.

⁵⁶ Soweit ersichtlich existieren keine unmittelbaren gesetzlichen Regelungen, die eine zwingende Verabschiedung von Gesetzen in deutscher, informeller Sprache vorschreiben. Da Gesetze jedoch insbesondere für die Bürger – zumindest in der Theorie – verständlich sein müssen und die Amtssprache in Deutschland deutsch ist (vgl. § 23 Abs. 1 VwVfG), scheint die Verabschiedung in deutscher, informeller Sprache aus verfassungsrechtlichen Gründen geboten.

3.1 Mögliche Lösungsvarianten

Für die digitale Umsetzung eines Gesetzes sind mehrere Varianten möglich abhängig davon (a) welchen Zusammenhang es zwischen den natürlich sprachlichen Gesetzen und Modellen gibt, (b) zu welchem Zeitpunkt die Modelle erstellt werden sollen und (c) welche Personen die DSL nutzen sollen.

3.1.1 Charakteristik von Gesetzestexten

Die juristische Fachsprache und Verwaltungssprache ist geprägt von langen Sätzen, nominalen Umschreibungen, der Personifizierung unbelebter Gegenstände oder Sachverhalte, Komplexität in der Darstellungsweise durch Derivationen und formelhafte Wendungen (Hansen-Schirra, S., Neumann, S. 2004). Betrachtet man die Formulierungen in unterschiedlichen Arten von Gesetzen, so existieren hier unterschiedliche Domänen-spezifische Termini, und es gibt Unterschiede in der generellen Struktur der Texte, bspw. existieren einerseits berechnungslastige Steuergesetze andererseits prozesslastige Verfahrensbeschreibungen.

Gesetze setzen sich sprachlich aus unterschiedlichen Komponenten zusammen, wie **Domänen-spezifischen Begriffen** (z.B. Fristen mit tempo-

ralen Aspekten, Bescheide oder Rechnungen), in Text ausgedrückte **Berechnungsvorschriften**, **logische Konstrukte** wie z.B. Wenn-Dann-Beziehungen (siehe auch (Kar et. al 2019)), **Vorgänge** im Sinne von Prozessen und Prozessschritten und **Referenzen** auf andere Komponenten oder Begriffe. Zudem gibt es Charakteristika hinsichtlich des **Aufbaus** von Gesetzestexten in Paragraphen, Absätze, Nummern und Sätzen (siehe Abb. 5). Der Vollständigkeit halber sei angemerkt, dass fehlende oder unspezifische Tatbestandsmerkmale im Gesetz in der Praxis regelmäßig die Konkretisierung durch Gerichte bedürfen.

Versucht man nun diese natürliche Sprache in formaler Sprache (Modellen) abzubilden (siehe Abb. 5), so muss man diese unterschiedlichen Charakteristiken darstellen können. Diese heterogenen Eigenschaften (z.B. Abbildung von Dynamik und Struktur) lassen sich durch Modelle durch unterschiedliche DSLs abbilden, die für die jeweiligen Eigenschaften gestaltet werden. Durch Kompositionsmethoden können diese unterschiedlichen Modelle dann jedoch gemeinsam dazu herangezogen werden, um die entsprechenden IT-Systeme zu generieren. Diese Designentscheidungen wie die DSLs gestalten sein müssen, werden jedoch nicht nur durch die Charakteristik des Gesetzes beeinflusst: wie, wer

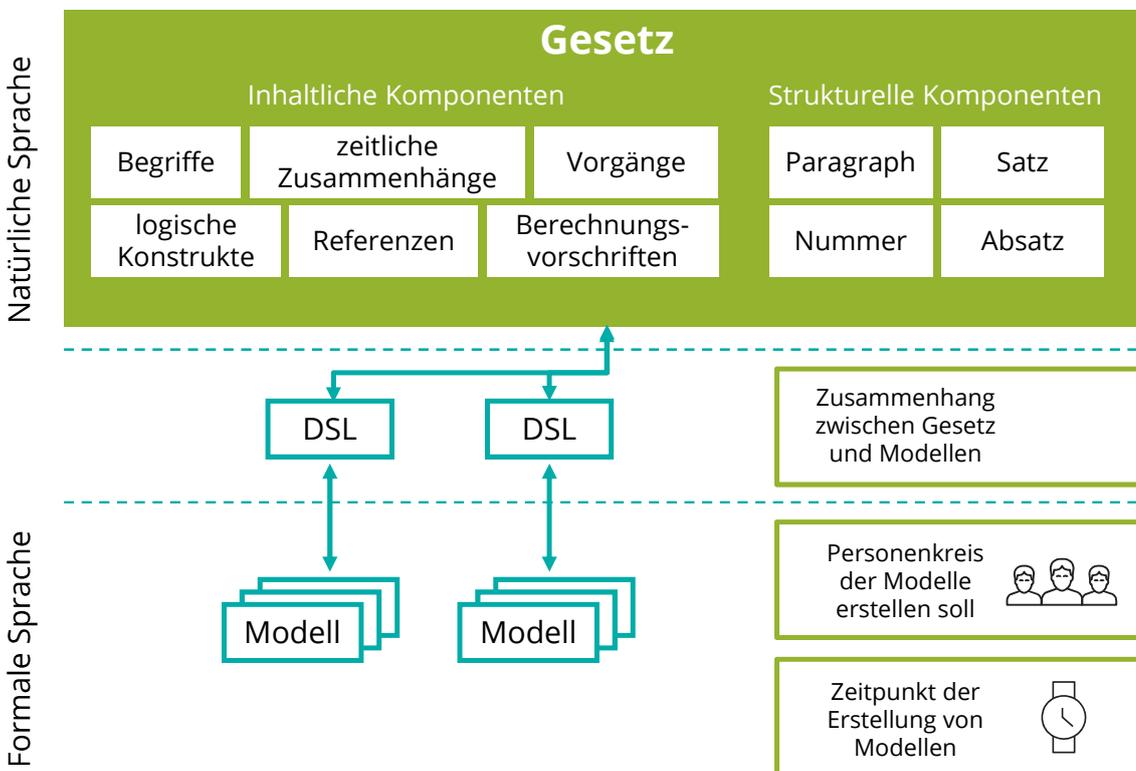


Abbildung 5: Zusammenhang zwischen natürlicher und formaler Sprache.

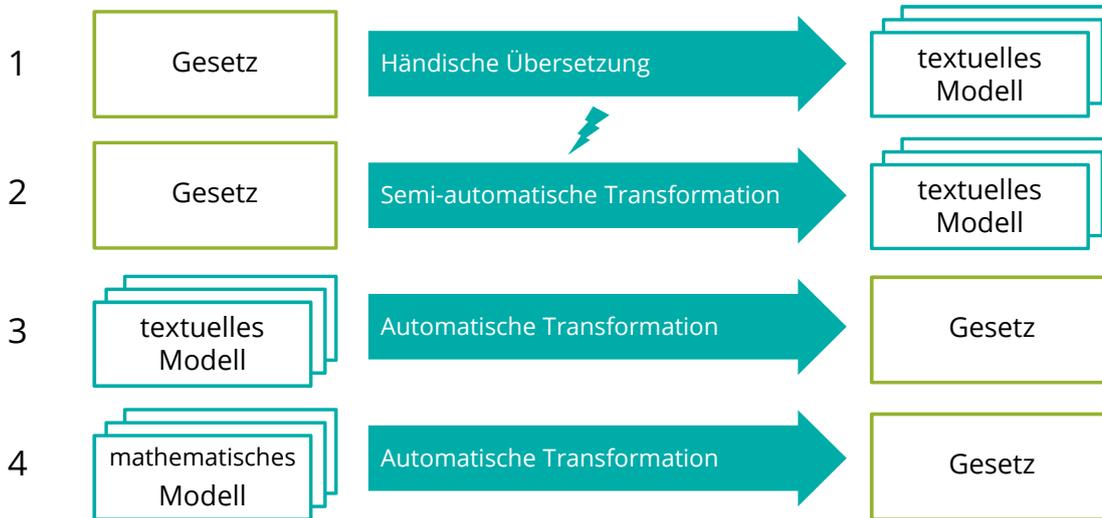


Abbildung 6:
Varianten des
Zusammenhangs
zwischen Gesetz
und Modell.

und wann diese Modelle erstellt bzw. erzeugt werden, hat ebenso einen starken Einfluss auf die Definition der DSL.

3.1.2 Der Zusammenhang zwischen Gesetzestexten und Modellen

Abhängig davon, welchen Zusammenhang und welche Abhängigkeiten man zwischen Gesetzestexten und IT-verarbeitbaren Modellen schaffen will, sind vier unterschiedliche Varianten der Gestaltung möglich (siehe Abb. 6): Jeweils natürlichsprachliche Beschreibung und dann händische Übersetzung oder semi-automatische Transformation in Modelle bzw. Beschreibung in textuellen oder mathematischen Modellen und dann jeweils Generierung des Gesetzestexts in natürlicher Sprache.

(1) *Gesetze werden natürlichsprachlich beschrieben und dann händisch in Modelle übersetzt.* Ein Vorteil besteht darin, dass die aktuellen Gesetzesentwurfsprozesse nicht stark verändert werden müssen. Ein großer Nachteil bei diesem Verfahren ist jedoch, dass durch die händischen Überführungen Fehler auftreten können bzw. dass es weiterhin Interpretationsspielräume geben kann, die vom Gesetzgeber nicht beabsichtigt wurden. Auch Ansätze, die den Text durch Annotationen verbessern wie z.B. (Off, T., Kühn, H., Schuppan 2016) sind nach wie vor nicht vor Fehlern sicher. Dieser Ansatz wird daher in der vorliegenden Studie nicht weiter betrachtet.

(2) *Gesetze werden natürlichsprachlich beschrieben, und dann semi-automatisch in formale Sprache transformiert.* Auch hier besteht der Vorteil, dass der Gesetzesentwurfsprozess

nicht verändert werden muss, da für die IT notwendige Modelle aus den natürlichsprachlichen Texten generiert werden. Ein großer Nachteil besteht hierbei jedoch darin, dass diese Transformation nicht voll automatisch durchgeführt werden kann. Durch manuelle Übersetzungsschritte können wiederum unbeabsichtigt Fehler auftreten. Dieser Prozess kann dadurch verbessert werden, wenn in Gesetzestexten semi-strukturierte natürliche Sprache verwendet werden würde, d.h. nicht die volle Mächtigkeit der deutschen Sprache, mit ihrer Ambiguität, starker Verschachtelung, Homonymen und Synonymen, ausgeschöpft und dadurch die natürlichsprachliche Verarbeitung (Natural Language Processing) vereinfacht werden würde. Dieser Ansatz kommt damit jedoch bereits der nächsten Variante (3) sehr nahe, mit dem Unterschied, dass die juristische Fachsprache geändert werden müsste. Wir gehen davon aus, dass dieser Ansatz in der Praxis nicht gewünscht ist und gehen im weiteren Verlauf daher ebenfalls nicht näher auf diese Variante ein.

(3) *Gesetze werden durch textuelle Modelle beschrieben, die dann in natürliche Sprache transformiert werden.* Dies bietet den Vorteil, dass die Modelle und der beschlossene Gesetzestext aufeinander abgestimmt und konsistent sind. Durch Verwendung dieser Modelle für die Systemgenerierung kann sichergestellt werden, dass die Gesetzestexte mit den IT-Systemen übereinstimmen. Voraussetzung für diese Variante sind jedoch notwendige Kenntnisse in der Formulierung von Modellen, sodass Juristen von diesem Vorgehen zunächst überzeugt und entsprechend geschult werden müssten. Davon spricht auch (McKinsey & Company 2019, S. 19), wenn es darum geht Fähigkeiten aufzubauen und das

passende Team für solche Herausforderungen zu bilden. Durch die gemeinsame Entwicklung einer DSL durch Juristen, Verwaltungsexperten und IT-Fachleuten kann man jedoch eine Sprache entwickeln, die der natürlichen Sprache sehr nahekommt und dadurch besser einsetzbar ist. Im Folgenden wird dieser Ansatz in der vorliegenden Studie weiterverfolgt.

(4) *Gesetze werden in mathematischen Modellen beschrieben, die dann in natürliche Sprache transformiert werden.* Ein Vorteil besteht in dieser Variante darin, dass die Formulierungen für IT-Systeme am einfachsten zu interpretieren sind. Ein Nachteil liegt jedoch in der deutlich komplexeren Erstellung der Modelle, die weiterreichende Kenntnisse voraussetzt. Aus diesem Grund wird dieser Ansatz, auch wenn dieser für IT-Fachleute wohl die am besten verarbeitbare Variante darstellen würde, in der vorliegenden Studie nicht weiter behandelt.

3.1.3 Zeitpunkt der Erstellung der Modelle

Betrachtet man mögliche Zeitpunkte, wann Modelle erstellt werden können, so gibt es auch hier unterschiedliche Varianten: a) Zu Beginn des Gesetzgebungsprozesses, b) mit Erstellung der textuellen Form des Gesetzes (entwurfs), c) direkt nach Beschluss des Gesetzes oder d) erst unmittelbar vor der Umsetzung in ein IT-System. Verwendet man DSLs, um Gesetze als Modelle zu formulieren, so startet der Prozess der Digitalisierung nach Variante a) oder zumindest nach Variante b).

Wir empfehlen hier möglichst frühzeitig anzusetzen, d.h. die Modelle selbst bereits im Rahmen des Gesetzgebungsprozesses (Variante a) oder Variante b)) zu erstellen und zu verwenden. So können diese ggf. gemeinsam mit der generierten natürlichsprachlichen Fassung der Gesetze beschlossen werden.

3.1.4 Prozessschritte und Nutzergruppen

Gesetze entstehen im Zusammenspiel verschiedener Beteiligten und werden auf unterschiedlichen Ebenen (Bund, Land, Kommune) erarbeitet, bearbeitet und beschlossen. Die praktische Anwendung der Gesetze in IT-Systemen betrifft wiederum unterschiedliche Ebenen der Verwaltung, die Bürgerinnen und Bürger sowie unterschiedliche andere Berufsgruppen wie z.B. Steuerberater. Verwendet man DSLs, so gibt es zwei voneinander weitgehend unabhängige Schritte: Die *Erstellung* und die *Nutzung der DSLs*.

Im **Entwurfsprozess einer DSL** (siehe Abb. 7), auch Language Engineering genannt, werden bestehende Gesetzestexte analysiert und als Beispiel herangezogen, um die Grammatik der DSL zu beschreiben. Die Grammatik beschreibt, welche sprachlichen Konstrukte in den Modellen verwendet werden können. Zudem können in diesem Schritt relevante Einschränkungen festgelegt werden. Dieser Prozess kann wiederholt werden, wenn unterschiedliche DSLs notwendig werden: z.B. eine DSL zur Beschreibung von Konzepten oder eine weitere DSL zur Definition

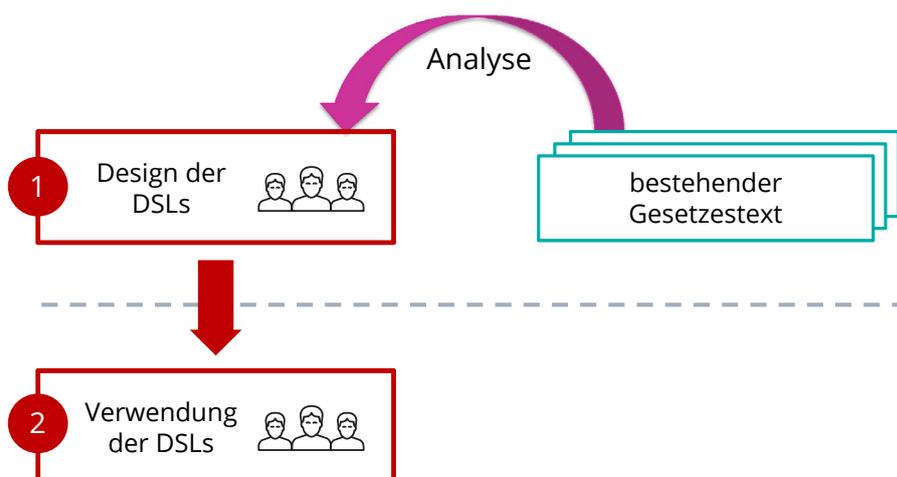


Abbildung 7: Prozess der Erstellung der DSLs.

von Prozessen. Diese DSLs können dann verwendet werden, um Modelle zu erstellen.

Die Personengruppe, die diese DSLs erstellt (siehe Gruppe 1 in Abb. 7), sollte sich aus IT-Experten, Verwaltungsexperten und Juristen zusammensetzen, um hier sowohl die technische, juristische als auch die Anwender-Expertise einbringen zu können.

Grundsätzlich ist es möglich DSLs zu verändern bzw. zu erweitern, wenn festgestellt wird, dass die Modelle nicht alle intendierten Aspekte ausdrücken können. Änderungen sollten jedoch nicht zu oft erfolgen, um bereits bestehende Modelle valide zu halten. Erweiterungen, die die alten Modelle nicht invalide machen, stellen jedoch kein Problem dar und können so schrittweise auch bereits bestehende Modelle verbessern.

Der Prozess der Entwicklung und Nutzung von Modellen (siehe Abb. 8) umfasst mehrere Schritte: Die Entwicklung bzw. Anpassung von Modellen, die Generierung des Gesetzestextes (in natürlicher Sprache) aus den Modellen, nach der erfolgten Beschlussfassung des Gesetzes die Generierung von Code und Implementierung notwendiger Schnittstellen zum bestehenden System, die Einbindung des Codes in die Anwendung sowie die Nutzung der Anwendung

z.B. für die Umsetzung in der Verwaltung, also Prüfung, Berechnung, politische Nutzung für „Modellrechnungen“, Prognosen (z.B. Steuerprognosen) sowie durch die Bürgerinnen und Bürger selbst.

Durch die Menge an Gesetzen, die auf unterschiedlichen Ebenen erarbeitet und beschlossen werden, ergibt sich, dass der Erstellungsprozess der Modelle für diese Menge ausgelegt sein muss. Betrachtet man beispielsweise die Statistik zur Gesetzgebung, so sind im Zeitraum 1990-2017 jedes Jahr im Durchschnitt über 130 Gesetzesvorhaben vom Bundestag verabschiedet worden (Deutscher Bundestag 2018), hinzu kommen noch die Gesetzesvorhaben von Ländern und Kommunen. Daraus ergibt sich aber auch, dass die Personengruppe, die diese DSLs verwendet, um Modelle zu erstellen (siehe Gruppe 2 in Abb. 7 und 8), sich aus anderen Personen zusammensetzt als diejenigen, die die DSL erstellt hat, da sie auf die unterschiedlichen politischen bzw. verwaltungstechnischen Ebenen verteilt sind. Es ist wichtig alle Nutzergruppen in die Gesetzesvorbereitung einzubinden, da eine Zentralisierung nicht möglich erscheint. (McKinsey & Company 2019, S. 19) schlägt für solch eine Einbindung z.B. die Implementierung von Gesetzgebungslaboren vor. In ebendiesen könnten auch die genannten Modelle gemeinschaftlich entwickelt werden.

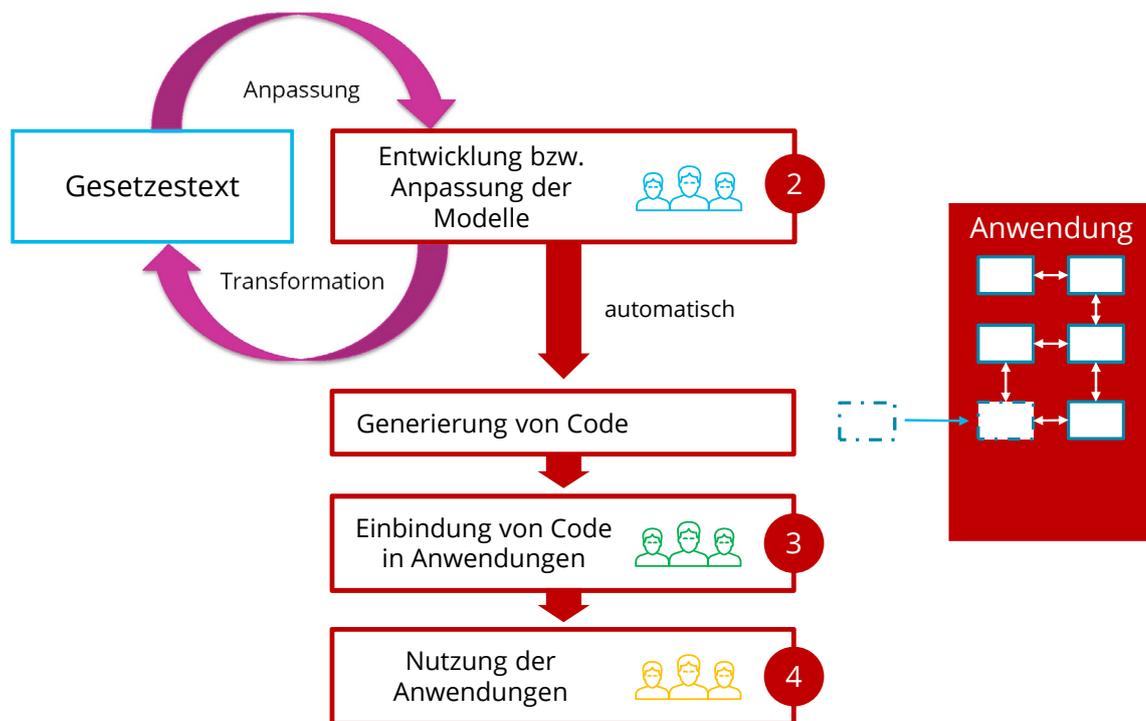


Abbildung 8: Prozess der Entwicklung und Nutzung von Modellen.

Die Personengruppen, die den generierten Code in bestehende Anwendungen integrieren und bei Bedarf händisch erweitern müssen (siehe Gruppe 3 in Abb. 8) sind IT-Fachleute. Auch diese können dezentral von unterschiedlichen Verwaltungseinheiten bzw. von externen Unternehmen aus agieren. Die Nutzerinnen und Nutzer der Anwendungen (siehe Gruppe 4 in Abb. 8) können unterschiedlichen Bereichen und Berufsgruppen angehören: Mitarbeiterinnen und Mitarbeiter in der Verwaltung, Bürgerinnen und Bürger oder Berufsgruppen wie z.B. Steuerberater.

3.1.5 Tooling und Interpretation

Jede DSL benötigt entsprechendes Tooling und Werkzeuge, um die Modelle sowohl bearbeiten als auch weiter verarbeiten zu können (France und Rumpe 2007), wie z.B. für die Analyse der Korrektheit der Modelle, Generatoren um Code zu erzeugen oder Tools um Simulationen durchführen zu können. Durch die Verwendung von Frameworks (eng. Language Workbenches) für die Entwicklung von Domänen-spezifischen Sprachen wie z.B. MontiCore, GEMOC Studio, Xtext oder anderen Frameworks sind damit verbundene Anwendungen einfacher zu realisieren.

Wie bereits zuvor beschrieben, können die Modelle verwendet werden, um z.B. Software-Code für Anwendungen zu generieren, Simulationen durchzuführen oder um zur Laufzeit einer Anwendung den Prozess zu steuern. Beispielsweise kann man fachliche Prozesse beschreiben, die ablaufen müssen um eine Steuer zu berechnen und definieren, wann Benutzerinteraktion seitens der Verwaltung notwendig ist. Es kann aber auch festgelegt werden, dass beispielsweise zu bestimmten Zeitpunkten Bescheide erlassen werden.

Ein anderer Anwendungsfall für die Modelle ist die Simulation z.B. mit statistischen Werten, d.h. zur Analyse von Gesetzesänderungen oder deren Auswirkungen. Hierfür kann man basierend auf bisherigen Erkenntnissen konkrete Werte für die Nutzerinnen und Nutzer hinterlegen und damit steuerliche Veränderungen simulieren.

Im Folgenden betrachten wir zwei konkrete Beispiele, wie solche DSLs gestaltet sein können.

3.2 Eine DSL zur Abbildung von gesetzlichen Steuerberechnungen und Validierung

Im Bereich der Steuerverwaltung ist für die Plausibilisierung von Einträgen in Steuererklärungen bereits seit über 10 Jahren eine DSL im Einsatz. Mit Hilfe dieser DSL werden die Steuerdaten überprüft und zum Teil auch steuerliche Berechnungen vorgenommen.

Zu jeder Steuerart und jedem Veranlagungszeitraum wird ein Modell spezifiziert, das die Plausibilität der eingegebenen Daten prüft. Die Plausibilität wird mit Hilfe von Validierungsregeln formuliert, deren Bedingung in einer DSL (Validierungssprache) formuliert werden.

Die Validierungssprache ist eine spezialisierte Sprache für Validierungen und Berechnungen in komplexen Fachdomänen, deren Strukturen baumartig aufgebaut sind und auf jeder Ebene geschachtelte Wiederholungen verwenden können. Die Sprache wurde mit dem Ziel entworfen, dass sie von Fachleuten aus den verschiedenen Fachreferaten der Finanzverwaltung eigenständig und ohne Unterstützung von IT-Experten eingesetzt werden kann. Die Sprache verwendet direkt das zugrundeliegende Felder-Modell, welches im Folgenden erläutert wird.

3.2.1 Felder-Modell

Das zugrundeliegende Felder-Modell beschreibt alle Felder einer Steuererklärung in einer hierarchischen Struktur. Es gibt in einer Steuererklärung verschiedene Vordrucke (Formulare), die aus Gruppen und beliebig vielen geschachtelten Untergruppen bestehen können. Abbildung 9 zeigt eine Beispielinstantz für die die Einkommensteuererklärung 2020 und den Vordruck bzw. die Anlage „Kind“. Es handelt sich um einen Screenshot aus dem durch die Finanzverwaltung verwendeten Modellierungs-Werkzeug. Für jedes Kind werden mehrere Untergruppen aufgelistet (**Angaben zum Kind**, **Kindchaftsverhältnis zur steuerpflichtigen Person/Ehemann/Person A**, usw.) und unter anderem auch die Gruppe **Entlastungsbetrag für Alleinerziehende**. Diese Gruppe ist mehrfach wiederholbar und enthält wiederum eine wiederholbare Gruppe zur Angabe von möglichen volljährigen Personen im Haushalt für die (zeitweise) kein Anspruch auf Kindergeld oder Freibeträge bestand (siehe in

Abbildung 9:
Gruppen und
Untergruppen.

E-DESIGNER		P-DESIGNER		
KOMPLETT AUFKLAPPEN		KOMPLETT ZUKLAPPEN		
Z-Nr.	Überschrift / Drucktext	SB	Kz.	Name
	▼ #{vordrucknr}. Anlage Kind			Kind
4+	> Angaben zum Kind			Ang_Kind
10+	> Kindschaftsverhältnis zur steuerpflichtigen Person / Ehemann / Person A			K_Verh_A
10+	> Kindschaftsverhältnis zur Ehefrau / Person B			K_Verh_B
11+	> Kindschaftsverhältnis zu einer anderen Person			K_Verh_and_P
16+	> Angaben für ein volljähriges Kind			Ang_vollj_Ki
20+	> Angaben zur Erwerbstätigkeit eines volljährigen Kindes			Ang_Erw_Taet_vj_K
31+	> Beiträge zur inländischen Kranken- und Pflegeversicherung (Nicht in der Anlage Vorsorgeaufwand enthalten)			KV_PV
41+	> Beiträge zur ausländischen Kranken- und Pflegeversicherung (Nicht in der Anlage Vorsorgeaufwand enthalten)			KV_PV_ausl
43+	> Übertragung des Kinderfreibetrags / des Freibetrags für den Betreuungs- und Erziehungs- oder Ausbildungsbedarf			Ueb_KFB_FB_BEAA
	▼ Entlastungsbetrag für Alleinerziehende			EFA
49	Das Kind war mit mir in der gemeinsamen Wohnung gemeldet vom - bis:	36	142	E0503801
50	Für das Kind wurde mir Kindergeld ausgezahlt vom - bis:	36	144	E0503802
51	Außer mir war(en) in der gemeinsamen Wohnung eine / mehrere volljährige Person(en) gemeldet, für die (zeitweise) kein Anspruch auf Kindergeld oder Freibeträge für Kinder bestand:	36	146	E0503701
51	Falls ja: vom - bis	36	147	E0503803
52	Es bestand eine Haushaltsgemeinschaft mit mindestens einer weiteren volljährigen Person, für die (zeitweise) kein Anspruch auf Kindergeld oder Freibeträge für Kinder bestand:	36	149	E0503821
52	Falls ja: vom - bis	36	150	E0503804
	▼ 1. Person			Hh_Gem
53	Name, Vorname			E0503805
54	Verwandtschaftsverhältnis			E0503806
54	Beschäftigung / Tätigkeit			E0503807

der Abbildung 9 exemplarisch 1. Person). Das Felder-Modell beschreibt alle Eingaben, die vom Steuerpflichtigen getätigt werden können, und eventuelle zusätzliche Felder für Zwischenbe-

rechnungen, die dem Steuerpflichtigen angezeigt werden. Die Eingaben in den Feldern entsprechen den Daten, die geprüft werden müssen und die Grundlage der Steuerberechnung sind.⁵⁷

⁵⁷ Das Beispiel ist aus der Modellierung der Einkommensteuer, Veranlagungszeitraum 2020 entnommen und zeigt den Teilbereich des Vordrucks "Kind". Die allgemein bekannten Papierformulare werden ebenfalls aus dem Felder-Modell generiert.

3.2.2 Aufbau der Sprache

Mit Hilfe der Validierungssprache werden zum Prüfen und Berechnen eine Menge von Regeln spezifiziert. Jede Regel formuliert eine Validierung oder einen Berechnungsschritt. Für jede Regel wird eine Bedingung spezifiziert. Die Sprache bietet eine Reihe von vordefinierten Bedingungen, Operationen und Vergleichen, die mit den Junktoren *Und* und *Oder* der klassischen Aussagenlogik verknüpft werden.

Zur Veranschaulichung soll das folgende einfache Beispiel dienen, das auf dem Beispiel-Feldermodell des vorangegangenen Abschnitts basiert. Eine (einfache) Regel überprüft folgenden Sachverhalt: *Beim Entlastungsbetrag für Alleinerziehende wurde angegeben, dass in der gemeinsamen Wohnung weitere volljährige Personen gemeldet waren, der Zeitraum in dem dies der Fall war wurde jedoch nicht erklärt.*

Diese Regel wird folgendermaßen formuliert:

```
[Außer mir waren in der gemeinsamen Wohnung eine/mehrere volljährige Personen gemeldet, für die (zeitweise) kein Anspruch auf Kindergel bestand] == Wahr Und FeldNichtAngegeben (Falls ja: vom -bis)
```

Die Regel wird für jedes angegebene Kind und jeden Entlastungsbetrag geprüft.

Bei Verwendung von sowohl *Und* als auch *Oder* in einer Bedingung wird zwingend eine Klammerung verlangt, auch wenn dies im Sinne der Aussagenlogik nicht notwendig ist. In der Praxis hat sich jedoch gezeigt, dass die Junktoren-Präzedenz für Nicht-Mathematiker nicht eindeutig war und häufig falsch eingesetzt wurde. Daher wurde in diesen Fällen eine Klammerung vorgeschrieben, die dann keinerlei Fehlinterpretation zulässt.

Die Validierungssprache bietet keine eigenständige Negation. Stattdessen werden alle vordefinierten Bedingungen jeweils in positiver und in negativer Form angeboten. Dadurch wird erreicht, dass die Teilbedingungen einfacher und einheitlich strukturiert zusammengesetzt werden. Die Gesamtbedingungen werden somit besser lesbar und klarer. Dies entspricht eher einem Aufbau, der für einen Domänenexperten verständlich ist. Beispielsweise ist die negative

Form der Bedingung `FeldAngegeben(<Feldname>)` durch die Bedingung `FeldNichtAngegeben(<Feldname>)` gegeben.

Auch Quantoren der Prädikatenlogik werden nicht als formal logische Teile der Sprache angeboten, sondern implizit über Operationen sowie die Strukturen, auf denen die Bedingungen formuliert werden. Dadurch wird erreicht, dass die Bedingungen sich an den Formulierungen eines Domänenexperten orientieren und somit verständlicher werden. So kann beispielsweise der Name einer Gruppe in der Regel verwendet werden, wenn eine Bedingung für alle Felder oder mindestens ein Feld der Gruppe formuliert werden soll. Zudem gibt es die Möglichkeit alle Wiederholungen einer Gruppe zu bezeichnen oder eine Bedingung für jede Wiederholung der Gruppe einzeln zu evaluieren. Wenn alle Wiederholungen einer Gruppe in der Bedingung referenziert werden, können diese durch einen Filteroperator eingegrenzt werden. Die Sprache verbindet somit die Einfachheit der Aussagenlogik mit der Ausdruckstärke der Prädikatenlogik und ist dadurch geeignet, um direkt von Steuerexperten verwendet zu werden.

Auch das folgende Beispiel bezieht sich auf den Felderbaum des Abschnitts 3.2.1. Die Regel überprüft folgenden Sachverhalt: *Beim Entlastungsbetrag für Alleinerziehende wurden Angaben zu Personen getätigt, es wurde jedoch weder ein Zeitraum angegeben, in dem diese Personen in der gemeinsamen Wohnung gemeldet waren, noch wurde ein Zeitraum angegeben, in dem eine Haushaltsgemeinschaft mit diesen Personen bestand.*

Diese Regel wird folgendermaßen formuliert:

```
KeinFeldAngegeben (Falls ja: vom -bis (Zeile 51), Falls ja: vom -bis (Zeile 52)) Und MindestensEineGruppeAngegeben (Person*)
```

Die Verwendung des Stern-Operators (bei `Person*`) spezifiziert, dass alle Wiederholungen der Gruppe „Person“ gemeint sind. D.h. durch den Stern-Operator wird eine Mengenbildung spezifiziert.

Auch die Eingabe von Regeln wird durch ein Werkzeug unterstützt. Für eine Kurznotation werden die technischen Bezeichnungen der Felder und Gruppen verwendet (siehe rechte

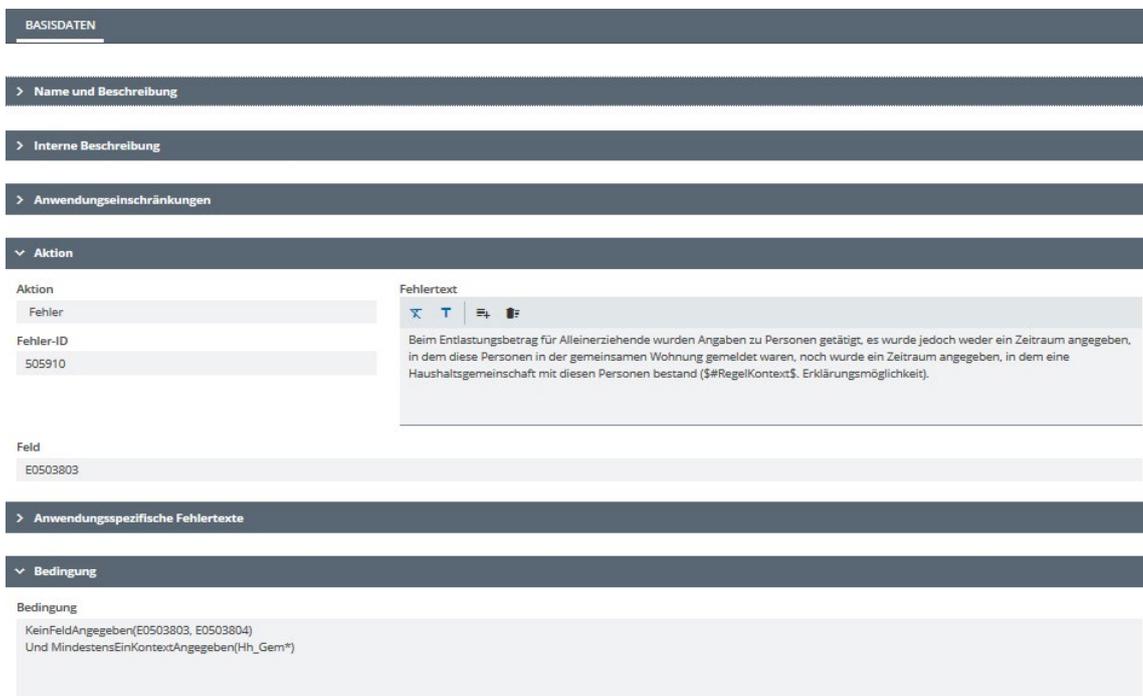


Abbildung 11:
Ansicht einer Regel
im Werkzeug.

Spalte im Felderbaum in Abb. 9). Abbildung 11 zeigt die Werkzeug-Sicht auf eine Regel.

3.2.3 Kurzüberblick über vorhandene Operatoren

Die Validierungssprache bietet eine Reihe von vordefinierten Bedingungs- und Berechnungsoperatoren an:

- Operatoren zum Prüfen, ob Felder bzw. Feldgruppen angegeben oder nicht angegeben sind: Beispielsweise `MindestensEinFeldAngegeben`, `MehrAlsEinFeldAngegeben`, `KeinFeldAngegeben`, `FelderNichtGemeinsamAngegeben`, `AlleFelderAngegeben` usw.
- Operatoren zum Prüfen von jeweils zwei Wertelisten: `WerteNichtEindeutig`, `MindestensEinFeldWertInWerteListe`, `KeinFeldWertInWerteListe`, `MehrAlsEinFeldWertInWerteListe`. Die Wertelisten können Konstanten oder wiederum andere Feldlisten sein.
- Operatoren zum Prüfen von Feldwerten, einfachen Berechnungen und Vergleichen.
- Operatoren auf Datumsangaben und Datumsbereichen, z.B. zum Prüfen, ob Datumsbereiche überlappend sind, Vergleich auf Vorher, Nachher, Vergleiche zum aktuellen Datum oder aktuellem Veranlagungszeitraum und Berechnung der Differenz zwischen Datumsangaben.

Im Folgenden wird ein Beispiel für Überprüfungen mit Datumsangaben gezeigt (aus der Einkommensteuer für den Veranlagungszeitraum 2020). Es wird geprüft, ob bei den Mehraufwendungen für doppelte Haushaltsführung für Mehraufwendungen für Verpflegung mehr Tage erklärt wurden, als sich aus den Angaben zur Dauer der doppelten Haushaltsführung ergeben. Dazu wird der Operator `DifferenzInTagen` verwendet, der zwischen zwei Datumsangaben die Differenz in Tagen ermittelt. Das Ergebnis wird verglichen mit der Summe mehrerer Eingabefelder, z.B.:

```
DifferenzInTagen (E0206103, E0206304)
+ 1 < Summe (E0207901, E0208011,
E0208103, E0208105)
```

Diese Regel wird automatisch für alle verschiedenen Angaben zur doppelten Haushaltsführung ausgewertet.

3.2.4 Notwendige Erweiterungen

Die Validierungssprache ist für Steuerexperten in den Fachreferaten der Finanzbehörden gut geeignet, um die Validierungs- und Berechnungsregeln einer Steuererklärung zu formulieren. Die Sprache bezieht sich immer ganz konkret auf die bereits genau spezifizierten Felder der jeweiligen Steuererklärung.

Für die Formulierung von Regeln im Rahmen eines Gesetzgebungsverfahrens sind die in der

Validierungssprache direkt referenzierten zu grundlegenden Felder-Modelle nicht unmittelbar geeignet, da sie bereits in dem für die IT-technische Umsetzung erforderlichen Detailgrad spezifiziert sein müssen – z.B. muss der konkrete Feldtyp (Zeichenkette, Zahl, Datum etc.) vollständig spezifiziert sein, d.h. z.B. mit maximaler Länge, dem Vorzeichen, dem maximalen Wert oder Datums-Formaten. Eine Sprache zur Prüfung im Rahmen der Gesetzgebung bzw. eines Gesetzgebungsverfahrens muss so erweitert werden, dass allgemeine Sachverhalte formuliert werden können ohne einen direkten Bezug auf die später zu spezifizierenden konkreten Felder herstellen zu müssen.

Im folgenden Abschnitt wird eine DSL zur Modellierung von gesetzlichen Fristen vorgeschlagen, die zusammen mit der DSL zur Berechnung und Validierung von Steuern verwendet werden kann. Dazu wird eine Reihe von allgemeinen Parameter modelliert, die zur Prüfung von gesetzlichen Fristen benötigt werden. Die Verwendung der modellierten Prüfungsvorgaben zu gesetzlichen Fristen für die Plausibilisierung z.B. einer spezifischen Steuerart erfordert somit ein Binding der vorgegebenen allgemeinen Parameter an die konkreten Felder einer Steuerart. Durch eine solche Verbindung und Konkretisierung wird es möglich, allgemeine Validierungsvorgaben direkt in konkrete Validierungsvorgaben einer Steuerart umzusetzen, z.B. wenn geprüft werden soll, ob eine Frist eingehalten wurde. Das Konkretisieren der allgemeinen Prüfungsvorgaben auf z.B. eine spezifische Steuerart erfordert eine entsprechende (manuelle) Modellierung dieser Verbindung. Durch dieses Verfahren wird ein Vorgehen nach einem Baukastenprinzip ermöglicht, in dem für die konkrete Verwendung jeweils auf höherwertige, allgemeiner formulierte Prüfungen zurückgegriffen werden kann.

3.3 Eine DSL zur Modellierung von gesetzlichen Fristen

Juristische Vorgänge unterliegen Fristen. Dieser Abschnitt diskutiert die Realisierbarkeit einer DSL zur Modellierung von gesetzlichen steuerlichen Fristen am Beispiel des Gesetzestextes zur Bestimmung der Festsetzungsverjährung in der AO (§§ 169-171 AO). Ist die Festsetzungsverjährung eingetreten, kann das Finanzamt einen Steuerbescheid nicht mehr erlassen, ändern oder aufheben.

Generell zielen Fristen darauf ab, das frühestmögliche und das spätestmögliche Datum zur Ausübung einer bestimmten Handlung bezüglich eines bestimmten Betreffs festzulegen. Die Fristen werden in der Regel vom Gesetzgeber festgelegt. Werden Handlungen getätigt, dann muss überprüft werden, ob diese fristgerecht getätigt wurden.

Die Frist zur Ausübung einer Handlung ist abhängig von einer Vielzahl von Parametern. Jeder Parameter entspricht einem Tatbestand, der die anzuwendende Frist bedingt. In Abhängigkeit von den Parametern muss eine obere Grenze und eine untere Grenze für die Zeitpunkte der Frist bestimmt werden, um überprüfen zu können, ob die Frist eingehalten wurde. Jedoch sind nicht immer alle Parameter bekannt. Trotzdem sollte auf Basis der bekannten Parameter entschieden werden, ob eine Frist eingehalten wurde, um Nachfragen, die den Prüfungsprozess unnötig verlängern würden, zu vermeiden.

In der bisherigen Praxis ist die Überprüfung der Einhaltung von Fristen weitestgehend eine manuelle Aufgabe, die nur für einfache Standardfälle mittels Computerunterstützung durchgeführt wird. Jedoch sind die Gesetzestexte zur Regelung der Fristen oft lang, kompliziert, und nicht intuitiv lesbar. Die Miteinbeziehung der hohen Anzahl der zu beachtenden Parameter, die die Frist beeinflussen, erschwert die manuelle Überprüfung der Einhaltung von Fristen zusätzlich.

Im Gegensatz zur manuellen Überprüfung wäre eine computerunterstützte Überprüfung eine leichte Aufgabe. Die Berechnungen der Fristen bestehen in der Regel aus grundlegenden einfachen Operationen und Vergleichen und sind daher unkompliziert und leicht automatisierbar.

Die Berechnungsvorschriften der Fristen könnten in einem Programm abgebildet werden. Die Überprüfung der Einhaltung einer Frist bestünde in diesem Fall aus einem Aufruf des Programms mit Werten für die Parameter. Das Programm würde automatisiert die Einhaltung der Frist überprüfen und könnte gegebenenfalls den Grund ausgeben, der verursacht, dass eine Frist nicht eingehalten wurde. Jedoch hat die direkte Abbildung der Berechnungen für Fristen in Programmen auch einige Nachteile. Domänenexperten ohne Programmierkenntnisse wären nicht dazu in der Lage, die Berechnungsvorschriften der Fristen im Programm zu ändern.

Der Gesetzestext würde getrennt vom Programmcode sein, was dazu führen würde, dass das Programm gegenüber dem Gesetzestext redundante Informationen darstelle. Das Programm könnte auch bezüglich des Gesetzestextes inkorrekt sein, was dazu führen würde, dass die Berechnungen inkonsistent gegenüber dem Gesetzestext wären.

Im Gegensatz zur direkten Abbildung der Berechnungsvorschriften in einem Programm würde eine explizite Modellierung der Fristen durch Modelle einer DSL weitere Vorteile mit sich bringen, sodass die Nachteile der direkten Abbildung in Programmcode nicht mehr bestehen würden. Die Modelle zur Modellierung der Fristen könnten als primäre Artefakte zur Regelung der Fristen eingesetzt werden. Die Modelle könnten direkt von Domänenexperten erstellt und verändert werden. Redundanzen und Inkonsistenzen vom Programmcode zum Gesetzestext würden damit nicht mehr entstehen können, da die Modelle als primäre Artefakte betrachtet und direkt von den Domänenexperten erstellt und verändert würden.

Regelungen zur Festsetzungsverjährung können auf verschiedene Arten interpretiert und modelliert werden. Es könnte intuitiv erscheinen, die Interpretation auf Basis einer Logik erfolgen zu lassen. Eine andere Möglichkeit ist eine Interpretation der Paragraphen, Absätze und Nummern als Sequenzen von Bedingungen, Zuweisungen und Anweisungen, ähnlich zu Programmen imperativer Programmiersprachen.

Im Folgenden werden diese beiden Interpretationsmöglichkeiten näher erläutert. Im nächsten Abschnitt 3.3.1 wird insbesondere näher auf die Festsetzungsfrist (§ 169 AO) eingegangen, welche im Folgenden als Beispiel eines Gesetzestextes zur Fristbestimmung verwendet wird. In Abschnitt 3.3.2 wird ein Datenmodell zur Berechnung von Fristen dargestellt. Der darauffolgende Abschnitt 3.3.3 beschreibt eine Interpretation und Modellierung auf Basis einer

Logik. Danach wird eine Interpretation und Modellierung auf Basis von Sequenzen von Bedingungen, Zuweisungen und Anweisungen, ähnlich zu Programmen imperativer Programmiersprachen, in Abschnitt 3.3.4 vorgestellt. Im Anschluss werden in Abschnitt 3.3.5 Modellierungsmöglichkeiten für Gesetzestexte zur Fristbestimmung aufgezeigt und beispielhaft Modelle aufgeführt. Schließlich werden im darauffolgenden Abschnitt 3.3.6 Aspekte der Automatisierung von Fristberechnungen und Fristvalidierungen durch Codegenerierung erläutert.

3.3.1 Festsetzungsverjährung, Fristen und Termine im Steuerrecht

Der Eintritt der Festsetzungsverjährung führt dazu, dass eine Steuerfestsetzung, deren Aufhebung oder Änderung (i.d.R. durch Bekanntgabe eines Steuerbescheids) nicht mehr zulässig ist.⁵⁸ Insbesondere das Vorliegen eines Sachverhaltes, der unter einen der Tatbestände der § 170 AO oder § 171 AO subsumiert werden kann, hemmt den Beginn oder den Ablauf der Festsetzungsfrist, sodass der Eintritt der Festsetzungsverjährung zeitlich in die Zukunft verlagert wird.

Insbesondere im Besteuerungsverfahren sind zahlreiche Fristen und Termine einzuhalten. Etwas eigenartig erscheint es daher, dass in der AO keine eigenen Definitionen dieser Begriffe vorhanden sind und grundsätzlich für die Berechnung von Fristen und für die Bestimmung von Terminen auf die §§ 187 bis 193 BGB verwiesen wird⁵⁹, wobei auch im BGB keine Definitionen dieser Begriffe zu finden sind. Diese Auffassung teilt offensichtlich auch die Finanzverwaltung und hat im Anwendungserlass zur Abgabenordnung (AEAO) die entsprechenden Definitionen aufgeführt.

So ist eine Frist ein abgegrenzter, bestimmter oder jedenfalls bestimmbarer Zeitraum.⁶⁰ Der Begriff Zeitraum ist allerdings nicht genauer definiert. Ein Termin ist hingegen ein bestimmter

58 Vgl. § 169 Abs. 1 Satz 1 AO.

59 Vgl. § 108 Abs. 1 AO.

60 Vgl. AEAO zu § 108 Nr. 1 Satz 1 mit Verweis auf BFH v. 14.10.2003, IX R 68/98, BStBl. II S. 898.

Zeitpunkt, an dem etwas geschehen soll oder zu dem eine Wirkung eintritt.⁶¹ Mit der Definition „Fälligkeitstermin“ wird auch das Verhältnis zwischen Fristen und Terminen klarer, indem dieser das Ende einer Frist angibt.⁶² In der steuerlichen Literatur finden sich weitere, konkretere Definitionen. So sei eine Frist ein abgegrenzter, bestimmbarer Zeitraum, vor dessen Ablauf eine Handlung oder ein Ereignis wirksam werden muss, um fristgerecht zu sein.⁶³

Die Festsetzungsfrist ist ein Zeitraum bis zu deren Ablauf die Steuerfestsetzung erfolgen, aufgehoben oder geändert werden kann.⁶⁴ Mit Ablauf der Festsetzungsfrist, was als Festsetzungsverjährung bezeichnet wird und ein Fälligkeitstermin ist, tritt die Wirkung ein, dass eine Festsetzung nicht mehr erfolgen, aufgehoben oder geändert werden kann⁶⁵ und, dass die Ansprüche aus dem Steuerschuldverhältnis insofern erlöschen können⁶⁶.

3.3.2 Datenmodell zur Berechnung von Fristen

Wie im vorangegangenen Abschnitt erläutert, werden die Begriffe Frist und Zeitraum weder in der AO noch im AEAO in einer Weise definiert, sodass sie direkt in einem Datenmodell darstellbar wären. Allerdings lassen sich aus der Festsetzungsverjährung (§§ 169-171 AO) die notwendigen Charakteristika einer Frist bezüglich des Zeitraums ableiten.

Der grundsätzliche Betrag der Frist wird als eine zeitliche Dauer in § 169 Abs. 2 AO festgelegt. Beispiele hierfür bilden die folgenden Auszüge: „Die Festsetzungsfrist beträgt: 1. ein Jahr für Verbrauchsteuern und Verbrauchsteuervergütungen“ (§ 169 Abs. 2 Satz 1 Nr. 1 AO), „vier Jahre für Steuern und Steuervergütungen, die keine Steuern oder Steuervergütungen im Sinne der Nummer 1 oder Einfuhr- und Ausfuhrabgaben

nach Artikel 5 Nummer 20 und 21 des Zollkodex der Union sind“ (§ 169 Abs. 2 Satz 1 Nr. 2 AO) und „Die Festsetzungsfrist beträgt zehn Jahre, soweit eine Steuer hinterzogen, und fünf Jahre, soweit sie leichtfertig verkürzt worden ist“ (§ 169 Abs. 2 Satz 2 AO).

Der § 170 AO verschiebt den Beginn der Frist in die Zukunft. Zu diesem Zweck definiert § 170 AO den regulären Beginn der Frist (Grundsatz) sowie Tatbestände, die den Beginn der Frist zeitlich nach hinten verlagern. Der Beginn einer Frist ist durch ein Datum definiert. Beispiele hierfür bilden die folgenden Auszüge: „Die Festsetzungsfrist beginnt mit Ablauf des Kalenderjahrs, in dem die Steuer entstanden ist oder eine bedingt entstandene Steuer unbedingt geworden ist“ (§ 170 Abs. 1 AO) und „[...] so beginnt die Frist [...] nicht vor Ablauf des Kalenderjahrs, in dem der Antrag gestellt wird“ (§ 170 Abs. 3 AO).

§ 171 AO trifft Aussagen zum Ende einer Frist. Das Erfüllen eines Tatbestands des § 171 AO führt nämlich in der Regel dazu, dass das reguläre Ende der Frist gehemmt wird. Das Ende einer Frist ist durch ein Datum definiert. Beispiele hierfür bilden die folgenden Auszüge: „[...] so endet die Festsetzungsfrist insoweit nicht vor Ablauf eines Jahres nach Bekanntgabe dieses Steuerbescheids“ (§ 171 Abs. 2 AO) und „[...] so endet die Festsetzungsfrist nicht vor Ablauf eines Jahres nach Eingang der Anzeige“ (§ 171 Abs. 9 AO).

Für die praktische Umsetzung eines Codegenerators zur Transformation von Modellen zur Modellierung der Fristenberechnung von gesetzlichen Fristen zu Quellcode sind allerdings unter anderem die folgenden Sonderfälle zu beachten: Fällt das Ende einer Frist auf einen Sonntag, einen gesetzlichen Feiertag oder einen Sonnabend, so endet die Frist mit dem Ablauf des

61 Vgl. AEAO zu § 108 Nr. 1 Satz 2.

62 Vgl. AEAO zu § 108 Nr. 1 Satz 3.

63 Vgl. Raub in Lexikonbeitrag aus Haufe Finance Office Premium, Fristen und Termine, Zusammenfassung.

64 Vgl. § 169 Abs. 1 Satz 1 AO „Umkehrschluss“.

65 Vgl. § 169 Abs. 1 Satz 1 AO.

66 Vgl. § 47 AO i.V.m. §§ 228ff. AO.

nächstfolgenden Werktags.⁶⁷ Ist für den Anfang einer Frist ein Ereignis oder ein in den Lauf eines Tages fallender Zeitpunkt maßgebend, so wird bei der Berechnung der Frist der Tag nicht mitgerechnet, in welchen das Ereignis oder der Zeitpunkt fällt.⁶⁸ Eine Frist, die nach Wochen, nach Monaten oder nach einem mehrere Monate umfassenden Zeitraum – Jahr, halbes Jahr, Vierteljahr – bestimmt ist, endet im Falle des § 187 Abs. 1 BGB mit dem Ablauf desjenigen Tages der letzten Woche oder des letzten Monats, welcher durch seine Benennung oder seine Zahl dem Tage entspricht, in den das Ereignis oder der Zeitpunkt fällt, im Falle des § 187 Abs. 2 BGB mit dem Ablauf desjenigen Tages der letzten Woche oder des letzten Monats, welcher dem Tage vorhergeht, der durch seine Benennung oder seine Zahl dem Anfangstag der Frist entspricht.⁶⁹ Fehlt bei einer nach Monaten bestimmten Frist in dem letzten Monat der für ihren Ablauf maßgebende Tag, so endet die Frist mit dem Ablauf des letzten Tages dieses Monats.⁷⁰

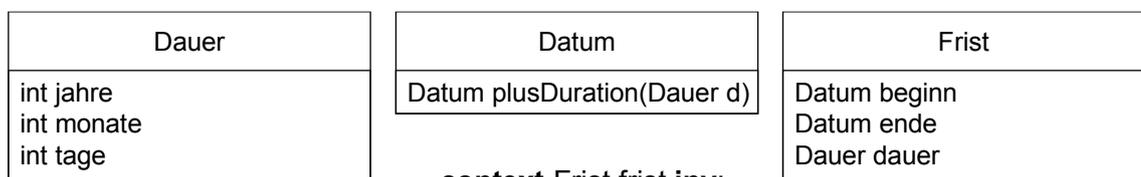
Insgesamt ergibt sich das in Abbildung 12 dargestellte Datenmodell zur Darstellung von Fristen. Eine Frist ist definiert durch einen Beginn, ein Ende und einen Betrag. Der Beginn und das Ende sind jeweils durch ein Datum definiert. Die Dauer einer Frist ist als eine zeitliche Dauer definiert. Das Ende einer Frist ist gleich dem Datum, das man erhält, wenn man ab Beginn der Frist die Dauer des Betrags der Frist verstreichen lässt. Durch die oben beschriebenen Sonderfälle wird ersichtlich, dass es sich bei der Methode `plusDuration(Dauer d)` der Klasse `Datum` nicht um eine generelle, sondern um eine spezifische, für juristische Fristen zugeschnittene Operation zur Behandlung von Daten handeln muss.

3.3.3 Interpretation auf Basis logischer Formeln

Zur Feststellung, ob eine Frist eingehalten wurde, müssen die betreffenden gesetzlichen Regelungen eingehalten werden. Intuitiv könnte es also naheliegend erscheinen, Gesetzesregelungen zur Bestimmung von Fristen als logische Formeln zu interpretieren und zu modellieren. Etwaige Nummern in Absätzen von Paragraphen könnten beispielsweise als logische Formeln modelliert werden. Die Komposition der Formeln der Nummern eines Absatzes durch logische Verknüpfungen würde dann eine Formel bilden, die den Sachverhalt des Absatzes modelliert. Die Komposition der Formeln der Absätze eines Paragraphen durch logische Verknüpfungen würde dann wiederum eine Formel bilden, die den Sachverhalt des Paragraphen modelliert. Analog könnten dann die Formeln verschiedener Paragraphen komponiert werden.

Nummern und Absätze würden dann in der Regel durch nicht-atomare Formeln repräsentiert werden, welche oft nur simple Implikationen beinhalten. In den Implikationen würde die Gegebenheit eines Umstands eine vom Umstand abhängige Fristveränderung implizieren. Die Gegebenheit des Umstands könnte als Wahrheitswert interpretiert werden. Die implizierte Fristveränderung hinge dann in der Regel von Daten ab, die mit dem Umstand assoziiert sind. Ob ein gegebenes Datum konform zur implizierten Fristveränderung ist, könnte durch einen Vergleich zweier Daten geprüft werden.

Jedoch beinhalten die Gesetzestexte in der Regel Verweise zwischen verschiedenen Absätzen, sodass eine kompositionale Auswertung der einzelnen Absätze nur schwer möglich ist. Die



context Frist frist inv:
`frist.ende.equals(frist.beginn.plusDuration(frist.dauer))`

Abbildung 12:
Datenmodell für
Fristen.

67 § 108 Abs. 3 AO.

68 § 187 Abs. 1 BGB.

69 § 188 Abs. 2 BGB.

70 § 188 Abs. 3 BGB.

Inhalte zur Festsetzungsverjährung sind eher in Form von Berechnungsvorschriften aufgebaut, die die Berechnung des Anfangs und des Endes einer Frist, abhängig von einer gegebenen Sachlage, ermöglichen. Daher sollten Gesetzestexte zur Festsetzungsverjährung nicht als logische Formeln interpretiert und somit auch nicht modelliert werden. Die folgenden Unterabschnitte detaillieren die obigen Erläuterungen.

3.3.3.1. Wertigkeit einer Logik zur Interpretation

Die Wertigkeit der zugrundeliegenden Logik zur Modellierung von Fristen könnte prinzipiell als zwei- oder dreiwertig angenommen werden.

Die Benutzung einer dreiwertigen Logik begründet sich darin, dass unbekannte Variablenwerte dazu führen könnten, dass beispielsweise für einen gegebenen Sachverhalt die Erfüllung von in Absätzen beschriebenen Tatbestände unbekannt sein könnten. Das heißt, dass jedem Absatz bezüglich der gegebenen Variablenbelegung ein echter Wahrheitswert (*wahr*, *falsch*) oder ein spezieller Wahrheitswert (*unbekannt*), der modelliert, dass der Wahrheitswert unbekannt ist, zugewiesen werden könnte. Im Kontext der Festsetzungsverjährung könnte beispielsweise unbekannt sein, ob „vor Ablauf der Festsetzungsfrist außerhalb eines Einspruchs- oder Klageverfahrens ein Antrag auf Steuerfestsetzung oder [...]“ (vgl. § 171 Abs. 3 AO) gestellt wurde. In diesem Fall wäre dann auch unbekannt, ob § 171 Abs. 3 AO durch die gegebene Variablenbelegung mit dem unbekanntem Wert erfüllt ist. Auch wenn grundsätzlich durch die Nutzung von DSLs und Modellen im Rahmen von Gesetzgebungsverfahren möglichst präzise Gesetze verabschiedet werden sollten, sei der Vollständigkeit halber erwähnt, dass die dreiwertige Logik auch bei unpräzise formulierten Regelungen angewendet werden kann. So ist beispielsweise strittig, welche genauen Tatbestände eine Ablaufhemmung nach § 171 Abs. 14 AO bewirken.⁷¹ Solange dies ungeklärt ist, könnte insoweit der spezielle Wahrheitswert (*unbekannt*) verwendet werden, ggf. mit besonderer Kennzeichnung, dass dies aufgrund von unpräzise formulierten Regelungen erfolgt. Ggf. könnte dieser Fall auch durch entsprechende

Modellierung wie in Abbildung 15 (siehe Abschnitt 3.3.5) erfolgen. Aus Vereinfachungsgründen wird der Fall der unpräzise formulierten Regelungen in dieser Studie nicht explizit behandelt.

Die Benutzung einer dreiwertigen Logik würde jedoch Interpretationsschwierigkeiten bezüglich der Komposition der drei Wahrheitswerte mit sich bringen. In manchen Kontexten wäre es sinnvoll, die Konjunktion so zu definieren, dass die Konjunktion vom Wert *wahr* mit dem Wert *unbekannt* den Wert *unbekannt* ergibt. Dies ist der Fall, wenn die Erfüllung eines Absatzes unbekannt ist, weil Variablenwerte *unbekannt* sind, und dieser Absatz zwangsläufig erfüllt sein müsste, damit die Frist eingehalten wurde. Wenn beispielsweise unbekannt ist, ob „vor Ablauf der Festsetzungsfrist außerhalb eines Einspruchs- oder Klageverfahrens ein Antrag auf Steuerfestsetzung oder [...]“ (vgl. § 171 Abs. 3 AO) gestellt wurde, dann wäre das durch § 171 Abs. 3 AO definierte Fristende unbekannt. Ein anderer Paragraph, ohne Einfluss auf das Fristende, könnte erfüllt sein. Da jedoch die Erfüllung des § 171 Abs. 3 AO unbekannt ist, kann nicht gefolgert werden, dass das Fristende eingehalten wurde, soweit das durch § 171 Abs. 3 AO definierte Fristende dafür relevant ist.

In anderen Kontexten wäre es sinnvoll, die Konjunktion so zu definieren, dass die Konjunktion vom Wert *wahr* mit dem Wert *unbekannt* den Wert *wahr* ergibt. Dies ist der Fall, wenn die Erfüllung eines Absatzes, unabhängig von der Erfüllung eines anderen Absatzes, die Einhaltung der Frist impliziert. Ist beispielsweise bekannt, dass „beim Erlass eines Steuerbescheids eine offenbare Unrichtigkeit unterlaufen“ ist (§ 171 Abs. 2 Satz 1 1. Hs. AO), während die Steuer innerhalb „eines Jahres nach Bekanntgabe dieses Steuerbescheids“ (§ 171 Abs. 2 Satz 1 2. Hs. AO) festgesetzt wurde, dann kann aus § 171 Abs. 2 Satz 1 AO bereits gefolgert werden, dass die Festsetzungsfrist noch nicht abgelaufen ist. Diese Folgerung ist unabhängig davon, ob auf Basis eines anderen Absatzes bekannt oder unbekannt ist, ob die Festsetzungsfrist bezüglich dieses anderen Absatzes (ggf. insoweit) abgelaufen ist.

⁷¹ Vgl. hierzu das zum Zeitpunkt der Erstellung der Studie anhängige Revisionsverfahren des BFH, V R 27/20.

Um falsche Folgerungen auszuschließen, sollte die Konjunktion im Kontext einer dreiwertigen Logik so definiert sein, dass die Konjunktion des Wahrheitswertes *wahr* mit dem Wahrheitswert *unbekannt* den Wert *unbekannt* ergibt. In diesem Fall würde dann aber keine Folgerung bezüglich der Erfüllung im letzteren Beispiel möglich sein, obwohl die Erfüllung des einen Absatzes schon die Einhaltung der Frist impliziert.

Durch die Benutzung einer zweiwertigen Logik wird obiger Problemfall per Konstruktion ausgeschlossen: Jeder Absatz wäre bezüglich der gegebenen Variablenbelegung entweder erfüllt oder nicht erfüllt. Jedoch wirkt es im Kontext einer zweiwertigen Logik herausfordernd, dass der Wahrheitswert aller repräsentierten Formeln auswertbar sein müsste, damit ein Wahrheitswert für die gesamte Formel ermittelt werden kann. Dafür müssten die Werte aller Variablen bekannt sein, damit für alle Formeln eine Auswertung erfolgen kann. In diesem Fall kann man die Belegung unbekannter Variablen durch sinnvolle Standardwerte (defaults) handhaben. Beispielsweise könnte ein unbekannter Wert einer Variable standardmäßig als *wahr* oder als *falsch* interpretiert werden. Der Standardwert für jede Variable könnte direkt vom Modellierer bestimmt werden.

3.3.3.2. *Komplikationen bei der Interpretation als logische Formeln*

Eine Komplikation der Interpretation als logische Formel besteht darin, dass alleine durch den Aufbau des Gesetzestextes nicht klar ist, mit welcher logischen Verknüpfung (Konjunktion, Disjunktion) die Formeln der einzelnen Paragraphen miteinander verknüpft werden müssen. Abhängig von der Formulierung zweier Absätze, müssen zur Einhaltung der Frist entweder die Formeln beider Absätze oder nur die Formel eines Absatzes erfüllt sein. Es ist daher nicht möglich, standardmäßig eine der beiden logischen Verknüpfungen für die Verknüpfung der Formeln, beispielsweise von Absätzen zu wählen. Würden Gesetze also als logische Formeln interpretiert und insbesondere auch formelartig modelliert werden, dann wäre das Explizieren der logischen Verknüpfungen der Absätze unabdingbar.

Ein Beispiel, bei dem die Komposition durch logische Disjunktion sinnvoll wäre, ist durch die zwei Absätze „Wird ein Steuerbescheid mit ei-

nem Einspruch oder einer Klage angefochten, so läuft die Festsetzungsfrist nicht ab, bevor über den Rechtsbehelf unanfechtbar entschieden ist; [...]“ (§ 171 Abs. 3a Satz 1 AO) und „Erstattet der Steuerpflichtige vor Ablauf der Festsetzungsfrist eine Anzeige nach den §§ 153, 371 und 378 Abs. 3 (AO), so endet die Festsetzungsfrist nicht vor Ablauf eines Jahres nach Eingang der Anzeige.“ (§ 171 Abs. 9 AO) gegeben. Die Verknüpfung der Absätze ist so zu interpretieren, dass bei der Erfüllung jedes individuellen Absatzes in einem bestimmten Zeitpunkt die Ablaufhemmung der Frist für diesen Zeitpunkt gegeben ist. Wenn der Endzeitpunkt der Frist bezüglich eines der beiden Absätze eingehalten wurde, dann wurde er auch generell eingehalten. Falls einer der beiden Absätze nicht erfüllt ist, bedeutet dies nicht zwangsläufig, dass die Frist nicht eingehalten wurde, da der andere Absatz noch erfüllt sein könnte.

Ein Beispiel bei dem die Komposition durch logische Konjunktion sinnvoll wäre, ist durch die zwei Absätze „Die Festsetzungsfrist beginnt mit Ablauf des Kalenderjahrs, in dem die Steuer entstanden ist oder eine bedingt entstandene Steuer unbedingt geworden ist“ (§ 170 Abs. 1 AO) und „Wird ein Steuerbescheid mit einem Einspruch oder einer Klage angefochten, so läuft die Festsetzungsfrist nicht ab, bevor über den Rechtsbehelf unanfechtbar entschieden ist [...]“ (§ 171 Abs. 3a Satz 1 AO) gegeben, wobei in diesem Beispiel von den anderen Absätzen abstrahiert wird. Die Verknüpfung der Absätze ist so zu interpretieren, dass die Erfüllung des Sachverhalts des einen Absatzes nicht impliziert, dass die Frist insgesamt eingehalten wurde. Es müssten also (unter Abstraktion der anderen Absätze der Festsetzungsfrist) die Sachverhalte beider Absätze erfüllt sein, damit die Frist eingehalten wurde. Dies ist darin zu begründen, dass der eine Absatz eine Aussage über den Beginn der Frist trifft, während der andere Absatz eine Aussage über das Ende der Frist trifft. Nur weil der Beginn einer Frist erfolgt ist, ist nicht zwangsläufig auch das Ende einer Frist erfolgt. Umgekehrt gilt, dass nicht zwangsläufig der Beginn einer Frist erfolgt, weil das Ende der Frist erfolgt ist.

Eine weitere Komplikation der Interpretation auf Basis von logischen Formeln ist, dass schon existierende Gesetzestexte nicht ohne weiteres als Formel interpretiert und modelliert werden können. Dies liegt unter anderem daran, dass existierende Gesetzestexte nicht hinsichtlich

des Grundprinzips der Kompositionalität (Hodges 2001) auf Formeln und ihren Verknüpfungen erstellt wurden. Das Prinzip der Kompositionalität fordert, dass die Bedeutung der Konjunktion zweier Formeln durch eine Komposition der Bedeutungen der einzelnen Formeln konstruiert werden kann. Die Kompositionalität geht dann verloren, wenn Inhalte eines Absatzes durch Inhalte eines anderen Absatzes außer Kraft gesetzt werden. Ein konkretes Beispiel bilden die Absätze 1 und 2 des § 170 AO. § 170 Abs. 2 AO setzt den Inhalt von § 170 Abs. 1 AO unter bestimmten Bedingungen außer Kraft („Abweichend von Absatz 1 beginnt die Festsetzungsfrist, wenn [...]“ (vgl. § 170 Abs. 2 Satz 1 AO)). Unter der Annahme, dass die Menge der Bedeutungen von Paragraphen, Absätzen und Nummern die Menge aller Abbildung von Variablenzuweisungen auf einen der zwei Wahrheitswerte *wahr* und *falsch* ist, kann Kompositionalität im obigen Fall nicht gegeben sein. Würde sie gegeben sein, dann könnte anhand einer Bedeutung festgestellt werden, welcher Paragraph die Bedeutung verursacht. Daher müssten Modelle zur Modellierung von Fristen zunächst einer Transformation unterzogen werden, um die kompositionale Form einer logischen Formel zu erhalten. Bezüge innerhalb der Sätze haben also zur Folge, dass beispielsweise Auswertungen der Formeln eines Absatzes von anderen Formeln anderer Sätze und deren Wahrheitswerten abhängig sein müssten.

3.3.4 Interpretation auf Basis von Bedingungen, Zuweisungen und Anweisungen

Dieser Abschnitt diskutiert eine Interpretation der Festsetzungsverjährung als Sequenz von Bedingungen und Anweisungen, ähnlich zu Programmen imperativer Programmiersprachen.

Es ist möglich, die Paragraphen, Absätze und Sätze als Sequenzen von Bedingungen und Anweisungen zu interpretieren. Insgesamt kann die Festsetzungsverjährung (§§ 169-171 AO) damit als eine Berechnungsvorschrift zur Berechnung des Beginns und des Endes von Fristen interpretiert werden.

Ein Gesetzestext besteht oftmals aus einer Vielzahl von Paragraphen. Ein Paragraph kann aus einem Satz oder aus mindestens zwei Absätzen mit jeweils mindestens einem Satz bestehen. Eine weitere Untergliederung durch Nummern und/oder Buchstaben ist möglich. Im Fall der Festsetzungsverjährung besteht der Gesetzes-

text aus Paragraphen, Absätzen und Nummern. Die Bedingungen und Anweisungen des Gesetzestextes ergeben sich aus der Komposition der Bedingungen und Anweisungen, die innerhalb der Paragraphen des Gesetzestextes definiert sind. Analog ergeben sich die Bedingungen und Anweisungen eines Paragraphen aus der Komposition der Bedingungen und Anweisungen, die innerhalb der Absätze des Paragraphen definiert sind, zusammen mit den Bedingungen und Anweisungen, die innerhalb des Paragraphen definiert sind. Wiederum ergeben sich die Bedingungen und Anweisungen der Absätze durch die Komposition der Bedingungen und Anweisungen, die innerhalb der Nummern des Absatzes definiert sind, zusammen mit den Bedingungen und Anweisungen, die innerhalb des Absatzes definiert sind. Nummern definieren eine Anweisungsreihenfolge von Bedingungen und Nummern, die nicht weiter dekomponiert sind.

Die Tatbestände, auf die innerhalb des Gesetzestextes verwiesen wird, sind Teil der Bedingungen und Anweisungen. In Abhängigkeit davon, ob ein Tatbestand erfüllt ist, können die jeweiligen Daten der Frist, abhängig vom Tatbestand, geändert werden. Die Tatbestände sind somit als Eingabeparameter zu interpretieren, die die Fristberechnung beeinflussen. In der AO sind die Typen dieser Eingabeparameter immer entweder Wahrheitswerte oder Datums. Da eine Vielzahl von Eingabeparametern definiert wird, sollte wie oben diskutiert, mit sinnvollen Default-Werten gearbeitet werden.

Jede Anweisung beeinflusst mindestens einen Bestandteil (Beginn, Ende, Betrag) der Frist. Jedoch beeinflusst nicht jede Anweisung alle Bestandteile der zu berechnenden Frist. Daher sind während der Berechnung der Frist nicht immer alle Bestandteile des Zwischenergebnisses der bisher berechneten Frist bekannt. Aus diesem Grund sollte ein Konzept **Partielle-Frist** eingeführt werden, das eine partiell berechnete Frist repräsentiert.

Anweisungen beeinflussen die Frist abhängig von ihrem bisherigen Wert und das Resultat einer Anweisung ist abhängig von der Belegung der Eingabeparameter, die die Tatbestände repräsentieren. Daher repräsentiert jede Anweisung eine Funktion, die eine partielle Frist (die bisher berechnete Frist) sowie die Belegung der Parameter der Tatbestände auf eine partielle Frist abbildet. Die entstehende partielle Frist

entspricht der neuen aktuell berechneten Frist in Abhängigkeit von der bisher berechneten Frist und den Parameterbelegungen, die die Tatbestände repräsentieren. Analog repräsentieren damit Paragraphen, Absätze und Nummern auch Funktionen mit den gleichen Definitionsbereichen und Wertebereichen.

Gesetzestexte können Verweise auf Paragraphen, Absätze und Nummern enthalten. Verweise dienen dazu, um die Berechnung einer partiellen Frist, die für einen Paragraphen, einen Absatz oder einer Nummer in Abhängigkeit vom Berechnungsergebnis der partiellen Frist eines anderen Paragraphen, eines anderen Absatzes oder einer anderen Nummer durchzuführen. Es ist also möglich, dass die Funktion eines Paragraphen, eines Absatzes oder einer Nummer zur Berechnung einer partiellen Frist abhängig von der Funktion eines anderen Paragraphen, eines anderen Absatzes oder einer anderen Nummer ist.

Um die folgenden Ausführungen simpel und damit leicht verständlich zu halten, vermindern wir ihre Komplexität, indem wir von Gesetzestextinhalten, die die Frist nicht vollumfänglich beeinflussen, abstrahieren. Diese Fälle können automatisch berücksichtigt werden, indem man zusätzlich zur vollumfänglichen Frist weitere Fristen berechnet, die nur für bestimmte Aspekte gelten. In der Regel können diese Fälle durch Schlüsselwörter erkannt werden. Beispielsweise beeinflusst „Ist beim Erlass eines Steuerbescheids eine offenbare Unrichtigkeit unterlaufen, so endet die Festsetzungsfrist insoweit nicht vor Ablauf eines Jahres nach Bekanntgabe dieses Steuerbescheids“ (§ 171 Abs. 2 Satz 1 AO) die Frist nur bezüglich der „offenbaren Unrichtigkeit“. In diesem Fall lässt das Schlüsselwort „insoweit“ erkennen, dass § 171 Abs. 2 Satz 1 AO die Frist nicht vollumfänglich beeinflusst.

3.3.5 Modellierung von Gesetzestexten zur Beschreibung von Fristen

Für jeden konkreten Sachverhalt sollten die einzuhaltenden gesetzlichen Fristen durch Modelle einer DSL modelliert werden können, um die Sprache zur Modellierung sämtlicher gesetzlicher Fristen, unabhängig von der konkreten Domäne des Gesetzestextes (z.B. Strafrecht, Steuerrecht), einsetzen zu können. Die DSL sollte natürlichsprachliche Konstrukte, welche der Jura-Domäne entnommen werden, zur Modellierung der Fristen in Abhängigkeit der zu prü-

fenden Variablen anbieten, um die DSL zugänglich für Domänenexperten zu gestalten.

Für die Umsetzung der Syntax der DSL gibt es zwei Extrem- und beliebig viele Mischformen. Die erste Extremform ist die Reduzierung der Gesetzestextinhalte auf das Notwendigste, um die Inhalte des Gesetzestextes bezüglich der Fristberechnung abbilden zu können. Die zweite Extremform besteht darin, die Syntax so zu gestalten, dass Gesetzestexte möglichst nah am ursprünglichen Gesetzestext modelliert werden können.

Die erste Extremform würde eine sehr künstliche Sprache implizieren, die ähnlich zu üblichen imperativen Programmiersprachen ist. Verschiedene Formulierungsvarianten könnten bis auf eine Menge notwendiger Varianten eingeschränkt werden, um möglichst eindeutige, präzise Formulierungen zu erzwingen. Beispielsweise könnte man einschränken, dass logische Aussagen nur durch die drei üblichen Junktoren (und, oder, nicht) modelliert werden können und dass logische Sachverhalte durch logische Ausdrücke zu realisieren sind. Mit der zweiten Extremform sind durch die Benutzung natürlicher Sprache durchaus auch Mischformen von Junktoren möglich, sodass logische Sachverhalte auch durch mehrere aufeinanderfolgende bzw. verschachtelte Sätze ausgedrückt werden können. Beide Extremformen können die zu modellierenden Sachverhalte gleich Akkurat abbilden. Jedoch sind die Formulierungen in der ersten Variante präziser. Dem gegenüber erlaubt die zweite Variante ausführlichere Beschreibungen, da sie natürlichsprachlich ist und somit für Domänenexperten und andere Personengruppen wohl intuitiv verständlicher wäre. Jedoch könnte die höhere Anzahl an logischen Operatoren zur Verknüpfung von Bedingungen, die die zweite Extremform durch die Benutzung natürlicher Sprache benötigt, dazu führen, dass durch falsche Kombinationen der logischen Operatoren eher fehlerhafte Modelle entstehen könnten. Daran angelehnt zeigt Abbildung 13 drei mögliche Varianten zur Modellierung des ersten Absatzes von § 170 AO.

In der ersten Variante werden die Gesetzesinhalte auf das notwendigste reduziert, um die Inhalte des Gesetzestextes bezüglich der Fristberechnung abzubilden. Als erstes werden der Paragraph und der Absatz angegeben. Im Absatz wird der Beginn der Frist durch `Beginn =` auf den Wert gesetzt, der auf der rechten Seite

Beispiel präzise Modellierung ohne verkürzte Parameternamen

```
§170  
(1) Beginn = [Ablauf des Kalenderjahrs, in dem die Steuer entstanden ist  
oder eine bedingt entstandene Steuer unbedingt geworden  
ist: Datum]
```

Beispiel präzise Modellierung mit verkürzten Parameternamen

```
Eingabeparameter:  
Datum steuerEntstandenOderUnbedingt  
/* Weitere Parameter, Paragraphen, Absätze, Nummern */  
§170  
(1) Beginn = steuerEntstandenOderUnbedingt
```

Beispiel natürlichsprachliche Modellierung

```
§170 Beginn Festsetzungsfrist  
(1) Die Festsetzungsfrist BEGINNT mit [Ablauf des Kalenderjahrs, in  
dem die Steuer entstanden ist oder eine bedingt entstandene Steuer  
unbedingt geworden ist: Datum].
```

Abbildung 13:
Drei Modellierungs-
möglichkeiten von
§ 170 Abs. 1 AO mit
expliziten Typ-
angaben.

der Zuweisung angegeben ist. In dieser Variante wird auf der rechten Seite ein Parameter definiert. Der Name bzw. die Beschreibung des Parameters und sein Wertebereich werden in den eckigen Klammern spezifiziert. In diesem Fall ist der Name des Parameters ausführlich wie im Absatz des Gesetzestextes angegeben. Nach dem Doppelpunkt in den eckigen Klammern folgt die Typangabe. In diesem Fall handelt es sich um ein Datum. Dieser sowie alle anderen modellierten Parameter dienen als Eingabe zur Fristberechnung.

Die zweite Variante ist ähnlich zur ersten Variante. Im Beispiel der zweiten Variante ist zusätzlich ein Parameterblock vorgesehen, bevor der Gesetzestext modelliert wird, in dem alle zu nutzenden Parameter definiert werden können oder müssen. In diesem Fall wird der Parameter `steuerEntstandenOderUnbedingt` mit dem Typ `Datum` definiert. Der Wert des Parameters soll das Datum repräsentieren, auf das im modellierten Absatz Bezug genommen wird. Dieser Parameter repräsentiert hier und im Folgenden sinngemäß genau den Ausdruck, der in der ersten Variante ausführlich ausgeschrieben ist, dies beinhaltet auch den „Ablauf des Kalenderjahrs“. Auf der rechten Seite der Zuweisung des Absatzes wird nun also nur noch der eingeführte Parameter referenziert. Während die erste Variante ausführlicher den Wert des Parameters beschreibt, ist die zweite Variante kompakter. In der zweiten Variante würden alle Parameter in einem Block vor dem modellierten Gesetzestext definiert werden, während die Parameter in der ersten Variante an den Stellen definiert werden,

an denen auf sie im Gesetzestext Bezug genommen wird.

Das Modell in der dritten Variante ist möglichst nah am Gesetzestext gestaltet. Der modellierte Wortlaut entspricht genau dem Wortlaut des Gesetzestextes. Das Ziel ist die Formulierungen des Gesetzestextes im Modell möglichst beibehalten zu können, einschließlich des Freitextes, der die Parameter näher erklärt. Ausschließlich der modellierte Operator (**BEGINNT**) ist in Großbuchstaben geschrieben, um ihn als expliziten Operator vom Wort *beginnt*, das im Freitext verwendet wird, unterscheiden zu können. Im Gegensatz zu den ersten beiden Varianten könnte die dritte Variante für Domänenexperten leichter verständlich sein, da sie näher am originalen Gesetzestext formuliert ist. Im Gegensatz dazu wird der Gesetzestext in den ersten beiden Varianten präziser definiert, was möglicherweise das Formulieren und Fehlerfinden in den Modellen erleichtern könnte.

Unter Bezugnahme auf die Ausführungen in Abschnitt 2.6.5 könnte die Verwendung von Modellen in der dritten Variante mit einer etwaigen verfassungsrechtlichen Verpflichtung zur Verabschiedung von Gesetzen in deutscher, informeller Sprache im Einklang stehen. Soweit eine solche Verpflichtung bestehen sollte, sollte ein besonderes Augenmerk auf die in den Abbildungen 13 bis 16 beispielhaft vorgestellten Varianten 3 gelegt werden.

Abgesehen von den beschriebenen drei Varianten bezüglich der Form des Modells des Geset-

Beispiel präzise Modellierung ohne verkürzte Parameternamen

```
§170
(1) Beginn = [Ablauf des Kalenderjahrs, in dem die Steuer entstanden ist
                oder eine bedingt entstandene Steuer unbedingt geworden
                ist]
```

Typangabe nicht notwendig

Beispiel präzise Modellierung mit verkürzten Parameternamen

```
Eingabeparameter:
steuerEntstandenOderUnbedingt
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
§170
(1) Beginn = steuerEntstandenOderUnbedingt
```

Typangabe nicht notwendig

Beispiel natürlichsprachliche Modellierung

```
§170 Beginn Festsetzungsfrist
(1) Die Festsetzungsfrist BEGINNT mit [Ablauf des Kalenderjahrs, in
    dem die Steuer entstanden ist oder eine bedingt entstandene Steuer
    unbedingt geworden ist].
```

Typangabe nicht notwendig

Abbildung 14:
Drei Modellierungs-
möglichkeiten von
§ 170 Abs. 1 AO
ohne explizite Typ-
angaben.

zestextes könnte man die Datentypen der Parameter auch automatisch inferieren. In diesem Fall müsste der Modellierer die Datentypen der Parameter nicht explizit angeben. Die Typen würden aus dem Kontext, in dem die Parameter benutzt werden, inferiert werden. Abbildung 14 zeigt die drei Varianten aus Abbildung 13 mit dem Unterschied, dass, auf explizite Typangaben verzichtet wird. Der Vorteil des Inferierens ist, dass die Formulierungen in den Modellen noch näher am Ursprungstext sind. Der Nachteil des Inferierens ist, dass die Datentypen von Parametern nicht mehr explizit an ihren Definitionsstellen sichtbar sind. Dies könnte Verständnisprobleme hervorrufen, die dazu führen, dass Modellierer nicht schnell erkennen, welchen Typ ein eingeführter Parameter hat. Die Modellierer müssten zunächst eine Benutzungsstelle des Parameters finden, um den jeweiligen Typen ableiten zu können. Alternative könnte auf Anfrage des Modellierers auch automatisch der Typ des Parameters berechnet und dem Modellierer präsentiert werden. Das Explizieren der Typangaben dient in diesem Fall nur der Erhöhung des Modellverständnisses der Modellierer und beeinflusst nicht die Weiterverarbeitungsmöglichkeiten des Modells. Ob explizite Typangaben oder das Inferieren von Typen gewünscht ist, sollte in Zusammenarbeit mit Domänenexperten entschieden werden, bevor die Sprache implementiert wird. In den folgenden Modellen beschränken wir uns auf die explizite Modellierung von Typangaben.

Der in Abbildung 13 modellierte Absatz 1 des § 170 AO ist leicht zu verstehen und zu modellieren. Die verschiedenen Modellierungsvarianten unterscheiden sich nur marginal. Im Gegensatz dazu sind andere Absätze des § 170 AO weniger leicht verständlich zu modellieren und die verschiedenen Modellierungsvarianten würden sich mehr unterscheiden. Abbildung 15 zeigt drei Varianten zur Modellierung des fünften Absatzes des § 170 AO. Dieser Absatz stellt verschachtelte Bedingungen dar, die dazu führen, dass für die Erbschaft- und Schenkungsteuer der Beginn der Frist abweichend vom Grundsatz in die Zukunft verlagert wird. Die Varianten sind nach dem gleichen Schema modelliert wie die Varianten, die in Abbildung 13 modelliert sind. In diesem Fall werden die Inhalte der drei Nummern des Absatzes nur berücksichtigt, wenn der Wahrheitswert, auf den am Anfang von Absatz 5 Bezug genommen wird, den Wert *wahr* hat. In den drei Nummern sind erneut Bedingungen über Parameterwerte modelliert. Falls einer der Parameter den Wert *wahr* hat, dann werden die Inhalte der zugehörigen Nummer berücksichtigt. Die Inhalte der Nummern spezifizieren, dass die Frist frühestens an einem bestimmten Datum beginnen darf. In den ersten beiden Varianten wird dies durch den Operator \geq abgebildet. Beispielsweise sagt der Ausdruck `Beginn >= ablaufKalenderjahr-FallSchenkung`, wie in der zweiten Variante modelliert, aus, dass die Frist frühestens an dem

Beispiel präzise Modellierung ohne verkürzte Parameternamen

```
/* Weitere Paragraphen, Absätze, Nummern */
§170
/* Absätze 1 bis 4 */
(5) wenn [Erbschaftsteuer (Schenkungssteuer): Wahrheitswert] dann
1. wenn [Erwerb von Todes wegen: Wahrheitswert] dann
    Beginn >= [Ablauf Kalenderjahr, in dem Erwerber Kenntnis Erwerb erlangt hat: Datum]
2. wenn [Schenkung: Wahrheitswert] dann
    Beginn >= [Ablauf Kalenderjahrs, in dem Schenker gestorben ist oder Finanzbehörde von der
vollzogenen Schenkung Kenntnis erlangt hat: Datum]
3. wenn [Zweckzuwendung unter Lebenden: Wahrheitswert] dann
    Beginn >= [Ablauf Kalenderjahrs, in dem die Verpflichtung erfüllt worden ist, Datum]
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
```

Abbildung 15:
Drei Modellierungs-
möglichkeiten von
§ 170 Abs. 5 AO.

Beispiel präzise Modellierung mit verkürzten Parameternamen

```
Eingabeparameter:
Wahrheitswert erbschaftsteuerSchenkungssteuer
Wahrheitswert erwerbTod
Datum ablaufKalenderjahrFallTod
Wahrheitswert schenkung
Datum ablaufKalenderjahrFallSchenkung
Wahrheitswert zweckzuwendungLebende
Datum ablaufKalenderjahrVerpflichtungErfüllt
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
§170
/* Absätze 1 bis 4 */
(5) wenn erbschaftsteuerSchenkungssteuer dann
1. wenn erwerbTod dann
    Beginn >= ablaufKalenderjahrFallTod
2. wenn schenkung dann
    Beginn >= ablaufKalenderjahrFallSchenkung
3. wenn zweckzuwendungLebende dann
    Beginn >= ablaufKalenderjahrVerpflichtungErfüllt
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
```

Beispiel natürlichsprachliche Modellierung

```
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
§ 170 Beginn der Festsetzungsfrist
/* Absätze 1 bis 4 */
(5) FÜR die [Erbschaftsteuer (Schenkungssteuer): Wahrheitswert] BEGINNT die Festsetzungsfrist
nach den Absätzen 1 oder 2
1. BEI einem [Erwerb von Todes wegen: Wahrheitswert] NICHT VOR [Ablauf des Kalenderjahrs, in
dem der Erwerber Kenntnis von dem Erwerb erlangt hat: Datum],
2. BEI einer [Schenkung: Wahrheitswert] NICHT VOR [Ablauf des Kalenderjahrs, in dem der
Schenker gestorben ist oder die Finanzbehörde von der vollzogenen Schenkung Kenntnis erlangt
hat: Datum],
3. BEI einer [Zweckzuwendung unter Lebenden: Wahrheitswert] NICHT VOR [Ablauf des
Kalenderjahrs, in dem die Verpflichtung erfüllt worden ist: Datum].
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
```

Datum beginnt, das in dem Parameter ablauf-KalenderjahrFallSchenkung gespeichert ist.

Bei der Modellierung von § 170 Abs. 5 AO wird Bezug auf sieben Parameter genommen. Die Nutzungs- und Definitionsstellen der Parameter sind in Abbildung 15 blau markiert. In der ersten und der dritten Variante werden die Parameter im Modell an den Stellen definiert, an denen sie genutzt werden. In der zweiten Variante werden die Parameter mit verkürzten Namen definiert, bevor der Gesetzestext modelliert wird. Dies führt dazu, dass das Modell für den Gesetzestext bei der zweiten Variante kürzer ist, als bei den anderen beiden Varianten. Im Gegensatz dazu könnten die Inhalte der Varianten eins und

drei für Domänenexperten leichter verständlich sein, da die Parameternamen näher an den Beschreibungen des Gesetzestextes sind.

Das Modell der dritten Variante in Abbildung 15 ist wieder möglichst nah am Gesetzestext modelliert. Der Wortlaut des Modells unterscheidet sich nicht vom Wortlaut des Gesetzestextes. Ausschließlich Operatoren sind in Großbuchstaben modelliert, um sie von den gleichen Wörtern, die im Gesetzestext benutzt werden, unterscheiden zu können.

Im Beispiel entspricht der Gesetzestext den schwarz eingefärbten Textinhalten, die nicht durchgehend in Großbuchstaben modelliert sind. Ein Beispiel ist der Modellinhalt "§ 170

Beispiel präzise Modellierung ohne verkürzte Parameternamen

```
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
§169
/* Absatz 1 */
(2)
1. wenn [Verbrauchssteuern: Wahrheitswert] oder
   [Verbrauchssteuervergütungen, Wahrheitswert] dann
   Betrag = 1 Jahr
2. wenn ([Steuern: Wahrheitswert] oder [Steuervergütungen: Wahrheitswert]) und
   ([nicht [Nummer 1: Bedingung] oder [Einfuhr- und Ausfuhrabgaben
   nach Artikel 5 Nummer 20 und 21 des Zollkodex der Union: Wahrheitswert]) dann
   Betrag = 4 Jahre
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
```

Abbildung 16:
Drei Modellierungs-
möglichkeiten von
§ 169 Abs. 2 Satz 1
Nr. 1 und 2 AO. § 169
Abs. 2 Satz 1 Nr. 2 AO
referenziert dabei auf
§ 169 Abs. 2 Satz 1
Nr. 1 AO.

Beispiel präzise Modellierung mit verkürzten Parameternamen

```
Eingabeparameter:
Wahrheitswert verbrauchsteuern
Wahrheitswert verbrauchsteuervergütungen
Wahrheitswert steuern
Wahrheitswert steuervergütungen
Wahrheitswert einfuhrAusfuhrabgabenUnion
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
§169
/* Absatz 1 */
(2)
1. wenn verbrauchsteuern oder verbrauchsteuervergütungen dann
   Betrag = 1 Jahr
2. wenn (steuern oder steuervergütungen) und
   ([nicht Bedingung(1.) oder einfuhrAusfuhrabgabenUnion) dann
   Betrag = 4 Jahre
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
```

Beispiel natürlichsprachliche Modellierung

```
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
§ 169 Festsetzungsfrist
/* Absatz 1 */
(2) Die Festsetzungsfrist BETRÄGT:
1. EIN JAHR FÜR
   [Verbrauchssteuern: Wahrheitswert] UND [Verbrauchssteuervergütungen: Wahrheitswert]
2. VIER JAHRE FÜR
   [Steuern: Wahrheitswert] UND [Steuervergütungen: Wahrheitswert], die KEINE Steuern oder
   Steuervergütungen im Sinne der [Nummer 1: Bedingung] ODER [Einfuhr- und Ausfuhrabgaben nach
   Artikel 5 Nummer 20 und 21 des Zollkodex der Union: Wahrheitswert] sind.
/* Weitere Parameter, Paragraphen, Absätze, Nummern */
```

Beginn der Festsetzungsfrist“. Dieser Freitextteil des Modellinhalts ist zur internen Weiterverarbeitung des Modells nicht relevant, da er inhaltlich nicht zur Fristberechnung beiträgt. In den ersten beiden Variante werden weniger Schlüsselwörter benutzt, als in der dritten Variante. So wird in den ersten beiden Varianten zur Einführung einer Bedingung ausschließlich das Schlüsselwort **wenn** verwendet, während in der dritten Variante für diesen Zweck die beiden Schlüsselwörter **FÜR** und **BEI** verwendet werden. Auf der einen Seite könnte dies Domänenexperten durch die Natürlichsprachlichkeit des Modells beim Verständnis helfen. Auf der anderen Seite könnte die Bereitstellung mehrerer verschiedener Schlüsselwörter für den gleichen Zweck zu Verständnisproblemen führen.

Regelungen in Gesetzestexten können Bezug auf andere Regelungen desselben oder eines anderen Gesetzes insbesondere durch Verweis

auf Paragraphen, Absätze oder Nummern nehmen. Beispielsweise nimmt § 170 Abs. 2 Satz 1 Nr.1 AO Bezug auf den Beginn der Frist wie er in § 170 Abs. 1 AO definiert wurde: „[...] es sei denn, dass die Festsetzungsfrist nach Absatz 1 später beginnt“ (vgl. § 170 Abs. 2 Satz 1 Nr. 1 AO). Analog können auch Bedingungen referenziert werden, die in anderen Paragraphen, Absätzen und Nummern definiert werden. Beispielsweise nimmt § 169 Abs. 2 Satz 1 Nr. 2 AO Bezug auf die Bedingung in § 169 Abs. 2 Satz 1 Nr. 1 AO: „[...] keine Steuern oder Steuervergütungen im Sinne der Nummer 1 [...]“ (vgl. § 169 Abs. 2 Satz 1 Nr. 2 AO). Abbildung 16 zeigt drei Varianten zur Modellierung von § 169 Abs. 2 Satz 1 Nr. 1 und 2 AO. Die drei Varianten sind schematisch gleich zu den Varianten der beiden Beispiele in den Abbildungen 14 und 15 aufgebaut. Die Verweise auf die Bedingung von § 169 Abs. 2 Satz 1 Nr. 1 AO sind jeweils in den Varianten in Abbildung 16 blau markiert. Bezüge auf

den Beginn und das Ende von Fristen wie sie in anderen Paragraphen, Absätzen und Nummern definiert werden, können ähnlich modelliert werden.

Neben den drei vorgestellten Varianten in den Beispielen sind eine Vielzahl von Mischformen möglich. Die vorgestellten Varianten stellen demnach lediglich die Extremformen zur Modellierung dar. Während jeweils die ersten beiden Varianten Modelle künstlich wirkender Sprachen darstellen, stellt jeweils die dritte Variante ein Modell dar, das natürlichsprachlich und nah am Gesetzestext modelliert ist. Um zu evaluieren, welche der Varianten am besten geeignet ist, sind zukünftig Miteinbeziehungen von Domänenexperten aus Personengruppen, die diese Sprachen einsetzen würden, notwendig.

3.3.6 Codegenerierung aus Gesetzestexten zur Beschreibung von Fristen

Unabhängig von der gewählten Variante zur Modellierung von Fristen, sollte zur automatischen Berechnung von Fristen eine Funktionalität bereitgestellt werden, um für eine gegebene Belegung der Parameter der Tatbestände die zugehörige Frist berechnen zu können. Die Berechnung der Frist sollte eindeutig sein. Die Funktionalität könnte somit insbesondere auch dazu genutzt werden, um zu überprüfen, ob eine Frist bezüglich einer Parameterbelegung eingehalten wurde. Die Funktionalität könnte aus den Modellen zur Modellierung der Frist in Form von Quellcode generiert werden oder die Modelle könnten interpretiert werden. Im Folgenden gehen wir näher auf den Aspekt der Codegenerierung ein. Auf ähnliche Art und Weise wäre die Umsetzung eines Interpreters möglich. Für die Codegenerierung legen wir eine objektorientierte Programmiersprache als Zielsprache zugrunde.

Eine Möglichkeit zur Codegenerierung besteht darin, aus jedem Modell eine Klasse zu generieren. Die im Modell definierten Parameter könnten zu Attributen der Klasse generiert werden. Der Konstruktor der Klasse könnte dann genutzt werden, um diese Parameter zu setzen. Die Klasse sollte eine Methode bereitstellen, um die Fristenberechnung in Abhängigkeit von den im Konstruktor gesetzten Parametern auszuführen.

Paragraphen, Absätze und Nummern können auf die Bestandteile der Berechnungsergebnisse für andere Paragraphen, Absätze und Num-

mern verweisen. Daher müssen die Berechnungsergebnisse für alle Paragraphen, Absätze und Nummern jeweils individuell referenzierbar sein. Analog können Verweise auf einzelne Bedingungen erfolgen. Aus diesem Grund müssen auch die Bedingungen der Paragraphen, Absätze und Nummern jeweils referenzierbar sein. Daher sollten für die Bedingungen und Berechnungen für die einzelnen Paragraphen, Absätze und Nummern jeweils individuelle Methoden generiert werden, um die jeweiligen Berechnungen individuell referenzieren zu können.

Die Berechnungen der Methoden der Bedingungen sind abhängig von den Attributen der Klasse. Die Methoden benötigen also keine weiteren Parameter und geben einen Wahrheitswert zurück. Die Berechnungen der Methoden, die für die Fristberechnung für die Paragraphen, Absätze und Nummern generiert werden, sind daher ebenfalls abhängig von den Attributen der Klasse und zusätzlich abhängig vom bisherigen Berechnungsergebnis. Daher sollten die Methoden jeweils einen Parameter vom Typ `PartielleFrist` haben. Analog sollte der Rückgabewert der Methoden ebenfalls vom Typ `PartielleFrist` sein.

Abbildung 17 zeigt beispielhaft einen Ausschnitt einer Klasse, die für ein vollständiges Modell der Festsetzungsverjährung (§§ 169-171 AO) generiert werden könnte. In Abbildung 17 orientieren sich die Namen der Attribute an den Namen die in den Beispielen aus Abschnitt 3.3.5 für die jeweils zweiten Variante benutzt wurden. Der Konstruktor erwartet einen Wert für jeden definierten Parameter. Die Methode berechnet die Frist in Abhängigkeit der Attribute der Klasse. Analog existieren Methoden für die Fristenberechnungen der einzelnen Paragraphen, Absätze und Nummern (z.B. `berechneParagraph170Absatz5Nummer1(PartielleFrist frist)`). Für die Bedingungen der Paragraphen, Absätze und Nummern werden jeweils Methoden generiert, die einzeln referenziert werden können (z.B. `bedingungParagraph170Absatz5Nummer1()`).

Die Rümpfe der Methoden für die Bedingungen in Abbildung 17 bilden die modellierten Bedingungen entsprechend ab. Die Rümpfe der Methoden für die Berechnungen bestehen für Paragraphen aus den Anweisungen, die im Paragraphen definiert sind, komponiert mit Methodenaufrufen der Methoden für die Absätze des Paragraphen. Die Reihenfolge der Anweisungen

Festsetzungsverjährung ...
Datum steuerEntstandenOderUnbedingt Boolean erbschaftSteuerSchenkungsteuer Boolean schenkung Boolean verbrauchsteuern Boolean verbrauchsteuervergütungen
<u>Festsetzungsverjährung(...)</u> PartielleFrist berechne() PartielleFrist berechneParagraph169(PartielleFrist frist) PartielleFrist berechneParagraph170(PartielleFrist frist) PartielleFrist berechneParagraph171(PartielleFrist frist) PartielleFrist berechneParagraph170Absatz5(PartielleFrist frist) PartielleFrist berechneParagraph170Absatz5Nummer1(PartielleFrist frist) Boolean bedingungParagraph170Absatz5Nummer1() Boolean bedingungParagraph170Absatz5Nummer2()

Abbildung 17:
 Beispielhafter Aus-
 schnitt einer Klasse,
 die für ein Modell,
 das die Festset-
 zungsverjährung
 modelliert, generiert
 werden könnte.

und Methodenaufrufe ist entsprechend der Reihenfolge im Gesetzestext aufgebaut. Analog können die Methodenrumpfe für Absätze und für die Methode `berechne()` konstruiert werden.

Abbildung 18 zeigt einen beispielhaften Ausschnitt eines Quellcodes, der für ein Modell der Festsetzungsverjährung (§§ 169-171 AO) generiert werden könnte. Ausschnitte aus möglichen Modellen, für die der Quellcode generiert werden könnte, sind in Abbildung 15 dargestellt. Die Methode `berechne()` (Z. 1-7) initialisiert zunächst ein neues `PartielleFrist` Objekt `frist` (Z. 1). Anschließend ruft die Methode die entsprechenden Methoden für die Berechnungen für die §§ 169, 170 und 171 AO auf, wobei jeweils die aktuell berechnete Frist übergeben wird. Nach der Durchführung der Berechnungen für die Paragraphen wird die Frist ausgegeben. Analog ruft die Methode zur Berechnung der Frist für § 170 AO (Z. 9-18) zunächst die Methoden für die Absätze von § 170 AO auf, wobei jeweils die aktuell berechnete Frist übergeben wird und der Wert der Frist jeweils zum Berechnungsergebnis der Methoden für die Absätze aktualisiert

wird (Z. 10-16) und gibt anschließend die berechnete Frist aus (Z. 17). Die Methode für die Berechnungen von § 170 Abs. 5 AO (Z. 20-27) überprüft zunächst, ob die Bedingung dieses Absatzes erfüllt ist (Z. 21). Falls dies der Fall ist, werden die Methoden für die Berechnungen der Nummern dieses Paragraphen aufgerufen (Z. 22-24), bevor die berechnete Frist zurückgegeben wird (Z. 26). Die Bedingung dieses Absatzes (Z. 29-31) gibt lediglich den Wert zurück, der im abgefragten Attribut gespeichert ist (Z. 30). Die Methode für § 170 Abs. 5 Nr. 2 AO (Z. 33-41) überprüft zunächst die Bedingung dieser Nummer (Z. 34). Falls die Bedingung erfüllt ist, dann wird der Beginn der Frist wie im Modell beschrieben aktualisiert. Falls der Beginn der Frist noch nicht angegeben ist (Z. 35), oder der aktuelle Beginn vor dem im Modell angegebenen Beginn liegt (Z. 36), dann wird der Beginn der Frist auf den im Modell angegebenen Beginn gesetzt (Z. 37). Bei der Implementierung des `setters` muss darauf geachtet werden, dass unter anderem die in Abschnitt 3.3.2 erläuterten Sonderfälle bezüglich des Datums berücksichtigt werden. Die Methode endet mit der Rückgabe der berechneten Frist (Z. 40).

Abbildung 18:
Beispielhafter Aus-
schnitt eines Quell-
codes, der für ein
Modell der Fest-
setzungsverjährung
(§§ 169-171 AO)
generiert werden
könnte.

```
01 public PartielleFrist berechne {
02     PartielleFrist frist = new PartielleFrist();
03     frist = berechneParagraph169(frist);
04     frist = berechneParagraph170(frist);
05     frist = berechneParagraph171(frist);
06     return frist;
07 }
08
09 public PartielleFrist berechneParagraph170(PartielleFrist frist) {
10     frist = berechneParagraph170Absatz1(frist);
11     frist = berechneParagraph170Absatz2(frist);
12     frist = berechneParagraph170Absatz3(frist);
13     frist = berechneParagraph170Absatz4(frist);
14     frist = berechneParagraph170Absatz5(frist);
15     frist = berechneParagraph170Absatz6(frist);
16     frist = berechneParagraph170Absatz7(frist);
17     return frist;
18 }
19
20 public PartielleFrist berechneParagraph170Absatz5(PartielleFrist frist) {
21     if(bedingungParagraph170Absatz5()) {
22         frist = berechneParagraph170Absatz5Nummer1(frist);
23         frist = berechneParagraph170Absatz5Nummer2(frist);
24         frist = berechneParagraph170Absatz5Nummer3(frist);
25     }
26     return frist;
27 }
28
29 public boolean bedingungParagraph170Absatz5() {
30     return this.erbschaftssteuerSchenkungssteuer;
31 }
32
33 public PartielleFrist berechneParagraph170Absatz5Nummer2(PartielleFrist frist) {
34     if(bedingungParagraph170Absatz5Nummer2()) {
35         if(!frist.getBeginn().isPresent() ||
36             frist.getBeginn().isBefore(ablaufKalenderJahrFallSchenkung)) {
37             frist.setBeginn(ablaufKalenderJahrFallSchenkung)
38         }
39     }
40     return frist;
41 }
```

3.4 Zusammenspiel von DSLs und unserem Ansatz

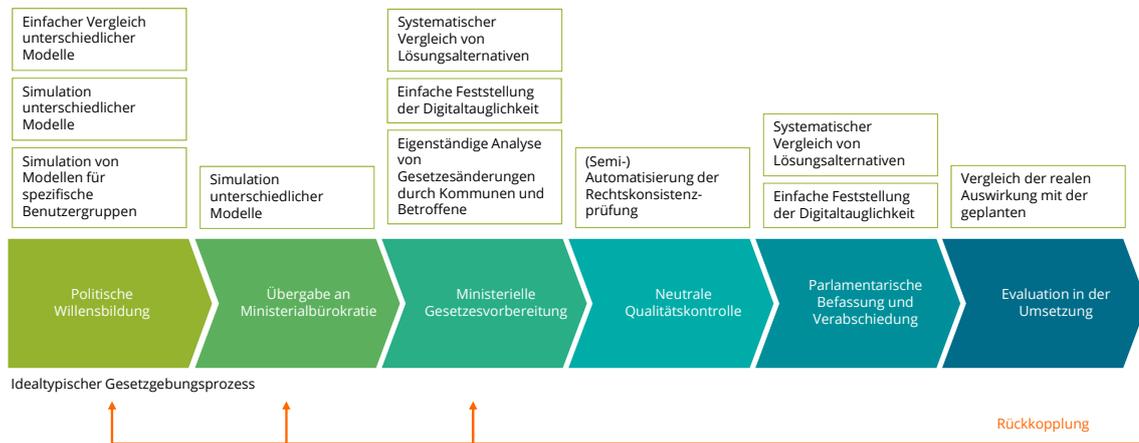
Die beiden beschriebenen DSLs haben gezeigt, dass die Nutzung von DSLs für die Beschreibung von Gesetzestexten durchaus praktikabel und handhabbar ist. Dennoch sei angemerkt, dass die Praktikabilität und Handhabbarkeit im Rahmen dieser Studie nur anhand von einigen Beispielen gezeigt wurde, woraus eine Allgemeingültigkeit nicht grundsätzlich ableitbar ist. Die Studienautoren der RWTH Aachen verfügen jedoch über umfassende Erfahrungen in der Erstellung von DSLs für unterschiedliche Anwendungsgebiete wie z.B. die Definition von Vertragsbestandteilen in einer stark natürlichsprachlichen DSL (Drave et. al 2020). Die Studienautoren der mgm technology partners gmbh verfügen über umfassende Erfahrungen in der praktischen Erstellung und Nutzung von DSLs für die Formulierung von steuerlichen Fachverfahren. Zusammen sind wir daher überzeugt, dass die Nutzung von DSLs für die Be-

schreibung von Gesetzestexten wie in dieser Kurzstudie skizziert, in der Praxis erfolgreich umgesetzt werden kann.

Wie bereits zuvor beschrieben, kann es notwendig sein, dass die Modelle aus unterschiedlichen DSLs zusammengenommen in einem IT-System realisiert werden sollten. Hierfür ist es notwendig, dass ein bzw. mehrere Generatoren Modelle unterschiedlicher DSLs als Input verwenden und daraus einen gemeinsamen Software-Code generieren. Die Komposition der Modelle der unterschiedlichen DSLs bzw. der Generatoren ist im Allgemeinen, insbesondere in der Praxis, eine große Herausforderung, die es noch zu lösen gilt.

Aus diesem Ansatz der digitalen Gesetzgebung ergeben sich eine Reihe von **Vorteilen** für die unterschiedlichen **Nutzergruppen**. Diese werden im Zusammenhang mit den im Gutachten im Auftrag des Normenkontrollrats (McKinsey & Company, Inc. 2019) skizzierten Schwächen

Abbildung 19:
Nutzung von
Modellen im Gesetz-
gebungsprozess.



im bisherigen Gesetzgebungsprozess in den folgenden Absätzen betrachtet. In Anlehnung daran zeigt Abbildung 19 einen Überblick über die Vorteile in den unterschiedlichen Prozessschritten eines Gesetzgebungsverfahrens unter Einsatz des in diesem Beitrag vorgestellten Ansatzes.

Gesetzgeber (Parteien und Parlamentarier)

- Erhöhung der Handlungsfähigkeit:** In Krisensituationen kann rasch gehandelt werden, da die Umsetzung von Änderungen in Gesetzen rasch in IT-Systemen realisiert werden kann.
- Vergleich unterschiedlicher Varianten von Gesetzesänderungen:** Werden z.B. durch Koalitionspartner oder auf unterschiedlichen Ebenen Vorschläge für Gesetzesänderungen erarbeitet, so können diese auf Modell-Ebene einfacher und automatisiert verglichen werden. Die Aushandlung bis zur beschlussreifen Vorlage kann durch die Nutzung von Modellen also zusätzlich unterstützt werden. Zudem muss man sich nicht bereits zu Beginn auf konkrete Maßnahmen festlegen, wenn unterschiedliche Modelle zu gleichen Simulationsergebnissen führen, sondern kann diese bis zur tatsächlichen Verabschiedung breiter diskutieren. Hierdurch kann das Defizit 1) „Beschränkung des Lösungsraums durch frühe Festlegung auf konkrete Maßnahmen“ in (McKinsey & Company, Inc. 2019) abgeschwächt werden.
- Simulation der Auswirkungen von Gesetzesänderungen:** Bereits bei der Willensbildung ist es möglich auf Modell-Ebene die Auswirkungen der Gesetzesänderungen zu simulieren, sowohl im Großen als auch im Kleinen. Im Großen ist hierfür z.B. der Rückgriff statistische Daten über Steuerpflichtige notwendig. Im Kleinen könnte man die Auswirkungen für bestimmte Zielgruppen z.B. mit exemplarischen Werten für alleinerziehende Erwerbstätige oder KMUs bis zu einer bestimmten Grenze berechnen. Dies ermöglicht es der Politik Auswirkungen besser abschätzen und Planspiele durchführen zu können.
- Einfache Feststellung der Digitaltauglichkeit:** Als Unterstützung für die Beschlussfassung kann die Digitaltauglichkeit von Gesetzen auch im Prozessschritt „Parlamentarische Befassung und Verabschiedung“ durch die Parlamentarier und Mitarbeiter analysiert werden.
- Systematischer Vergleich von Lösungsalternativen:** Politische Parteien haben durch die Nutzung von Modellen die Möglichkeit unterschiedliche Ansätze von unterschiedlichen Fraktionen direkt in ihren Sitzungen zu vergleichen und können (bei Bedarf) in einem Verhandlungsprozess selbst bis kurz vor der Beschlussfassung Anpassungen an ihren Gesetzesvorschlägen vornehmen.

- **Systematischer Vergleich von Lösungsalternativen:** Durch die Verwendung von Modellen können unterschiedliche Varianten einfacher untersucht und verglichen werden. Hierdurch kann das Defizit 2) „Wenig systematischer Vergleich von Lösungsalternativen“ in (McKinsey & Company, Inc. 2019) deutlich verbessert werden.
- **Einfache Feststellung der Digitaltauglichkeit:** Durch die Ausführung der Modelle können diese analysiert und eventuelle Fehler ausgegeben werden. Können sie erfolgreich durch die damit verbundenen Werkzeuge verarbeitet werden, so ist die Digitaltauglichkeit gegeben. Diese Überprüfungen können vor der tatsächlichen Beschlussfassung erfolgen und bei weiterführenden Änderungen einfach immer wieder durchgeführt werden. Hierdurch kann das Defizit 5) „fehlende systematische Prüfung der Digitaltauglichkeit“ in (McKinsey & Company, Inc. 2019) ausgeglichen werden.
- **(Semi-)Automatisierte Rechtskonsistenzprüfungen:** Die Ausführung der Modelle ermöglicht es diese auch mit anderen, bereits bestehenden Modellen abzugleichen. Bei einer entsprechend hohen Menge an zusammenhängenden Gesetzestexten, die alle bereits durch Modelle beschrieben sind, wären auch solche Rechtskonsistenzprüfungen einfacher durchzuführen.
- **Vergleich der geplanten mit realen Auswirkungen:** Durch Nutzung der Modelle und dafür entwickelte Analysetools können die Auswirkungen von Gesetzen anhand der real erhobenen Werte analysiert und mit den geplanten Werten verglichen werden.

- **Eigenständige Analyse von Gesetzesänderungen:** Im Gesetzgebungsprozess müssen kommunale Verbände eingebunden werden, wenn ein Gesetz Städte und Gemeinden betrifft. Wenn die definierten Modelle diesen Stellen zur Verfügung gestellt werden, um selbst die Auswirkungen analysieren zu können, ist so eine einfachere Einbindung und Interpretation der Auswirkungen möglich. Dies trägt etwas zu den Defiziten 3) „Unzureichende Sicht auf kommunalen Vollzug“ und 4) „Lückenhafte Beteiligung der Betroffenen“ in (McKinsey & Company, Inc. 2019) bei, kann diese Defizite jedoch nicht gänzlich ausgleichen.

Bürgerinnen und Bürger

- **Nachvollziehbarkeit von Gesetzesänderungen:** Dadurch, dass die Gesetze in Form von Modellen verabschiedet werden, sind diese auch für Bürgerinnen und Bürgern frei verfügbar. Bürgerinnen und Bürger hätten die Möglichkeit für sich selbst die Auswirkungen von Gesetzesänderungen zu simulieren, wenn entsprechende Tools zur Simulation zur Verfügung gestellt werden. Im Hinblick auf Defizit 6) „Geringe Nutzung von modernen Kollaborationsformen“ in (McKinsey & Company, Inc. 2019), wird hierbei auch von „hoher Komplexität der Sachfragen“ und „hoher Unsicherheit über Wirkungen“ gesprochen – diese Defizite können durch das entsprechende Tooling und Werkzeuge (Abschnitt 3.1.5) abgemildert werden. Dieser Ansatz erhöht somit auch die digitale Souveränität der Bürgerinnen und Bürger.

Zusammenfassend ist festzuhalten, dass sich eine Vielzahl an Möglichkeiten durch die Formulierung und Verabschiedung von Gesetzen in DSLs ergibt, die schlussendlich nicht nur die digitale Souveränität des Staates, sondern aller beteiligten Nutzergruppen erhöht.

4. HANDLUNGSEMPFEHLUNGEN FÜR DIE UMSETZUNG

Wie in dieser Studie gezeigt wurde, ist die Darstellung von Gesetzen durch DSLs und die automatisierte Auswertung von Modellen möglich. Wir haben dies anhand von Anwendungsbeispielen aus dem Steuerrecht gezeigt, ein analoges Vorgehen ist jedoch auch für Gesetzestexte aus unterschiedlichen Domänen z.B. Soziales, Umwelt oder Strafrecht durchführbar. Im Strafrecht könnte zur Begründungsunterstützung bspw. eine DSL entwickelt werden, die konkrete Ermessensspielräume als Intervalle, ggf. ergänzt um KI-basierte Durchschnittswerte, bei ähnlichen Fällen berechnet.

Es ist daher an der Zeit konkrete Projekte zu definieren und durchzuführen, in denen praktisch einsetzbare Werkzeuge entstehen. Konkret schlagen wir die folgenden Umsetzungsschritte vor:

- 1. Bildung interdisziplinärer Arbeitsgruppen zur Entwicklung von DSLs zur Abbildung von Gesetzestexten (organisatorisch und technisch):** Durch die Verwendung von DSLs wird die Digitalisierung von Gesetzen in vielen Bereichen erst ermöglicht bzw. in anderen beschleunigt, da sich die objektiv richtige Gesetzesanwendung direkt in Software übersetzen lässt. Zur Entwicklung von DSLs für die Formulierung von unterschiedlichen Gesetzestexten ist das Wissen von unterschiedlichen Beteiligten notwendig: Juristen, Verwaltungsexperten, Informatikern, etc. (siehe auch (Kar et al. 2019)). Diese Zusammensetzung muss sich für unterschiedliche Domänen z.B. Steuern, Soziales, Umwelt, etc. unterscheiden, da die jeweilige fachliche Expertise benötigt wird.
- 2. Entwicklung der Werkzeuge zur Verarbeitung und Code-Generierung aus Modellen (inhaltlich und technisch):** Für den praktischen Einsatz der DSLs müssen sich die relevanten Nutzergruppen darauf verständigen, welche IT Werkzeuge für die Modellierung eines Gesetzestexts notwendig sind und wie und durch wen diese Werkzeuge in einem weiteren Schritt entwickelt werden sollen.
- 3. Definition von DSLs und Modellen im Gesetzgebungsprozess, Generierung der natürlichsprachlichen Gesetzesfassungen und Generierung von Werkzeugen aus Modellen zur Unterstützung der Gesetzesanwendung (inhaltlich und technisch):** Diese Handlungsempfehlung zielt direkt auf die Steigerung der digitalen Souveränität des Staates ab, da die Gesetzgebung die Möglichkeit haben muss, möglichst präzise Gesetze formulieren zu können. Unter bestimmten Voraussetzungen plädieren die Studienautoren für die Verabschiedung von Modellen im Rahmen von Gesetzgebungsverfahren. Wir empfehlen konkrete Projekte umzusetzen. Dies kann die Arbeit an neuen Gesetzen, weiterführend aber auch an bestehenden Gesetzen sein.
- 4. Weiterbildung in der Verwaltung, von Juristen und politischen Entscheidungsträgern, um die Formulierung von Gesetzen in Modellen und deren Verständnis zu ermöglichen:** Es ist notwendig, die jeweiligen mit der Gesetzgebung befassten Ebenen darin weiter zu bilden, Gesetze in Modellen beschreiben zu können. Hierdurch liefert man einen kleinen Beitrag zum Ausgleich des Defizits 5) „Kein umfassendes Ausbildungskonzept für Legisten“ in (McKinsey & Company, Inc. 2019), welches allerdings über die hier formulierten Ansätze hinausgeht.
- 5. Anhebung von DSLs auf die Ebene von Standards und Normen:** DSLs, Modelle und Berechnungsvorschriften werden als Standards und Normen definiert, die sowohl in der zuständigen Behörde, als auch in der unternehmensinternen Software direkt zur Ausführung kommen können und daher zu einer wesentlich effizienteren IT und auch Zusammenarbeit der IT einzelner Verwaltungen und Unternehmen beitragen. Sie stellen damit einen essentiellen Teil der Digitalisierung dar.

Ogleich wir von den Vorteilen der von uns im Rahmen dieser Studie vorgeschlagenen Heran-

gehensweise, Gesetzestexte für sämtliche Gesetzgebungsverfahren mit Hilfe von DSLs zu formulieren, überzeugt sind, darf nicht unberücksichtigt bleiben, dass eine strikte Abkehr vom bisherigen Prozess der Gesetzgebung nicht praxistauglich erscheint. Um *Vorbehalte gegen den Einsatz von DSLs abzubauen*, sollte in Erwägung gezogen werden, die von uns vorgeschlagene Herangehensweise zunächst im Rahmen einzelner, ausgewählter Gesetzgebungsverfahren Anwendung finden zu lassen. Im direkten Vergleich zu den bisherigen Prozessen für andere Gesetzgebungsverfahren können die beteiligten Akteure *eigene Erfahrungen sammeln und sich selbst von den Vorteilen überzeugen*. Da die Nutzung von Modellen zu keinen Änderungen des grundsätzlichen Ablaufs und der Zuständigkeiten im Gesetzgebungsprozess, beispielsweise auf Bundesebene, wie in Abschnitt 2.6.2 beschrieben, führen sollte, sind die in diesem Abschnitt genannten Akteure zwingend mit einzubeziehen. Es sei an dieser Stelle auf die Handlungsempfehlungen für die Erstellung der Modell-Vorlage im Rahmen des Gesetzgebungsverfahrens in Abschnitt 2.6.2.6 verwiesen. Die Ausführungen in Abschnitt 2.6.5 sollten gleichfalls im Blick behalten werden.

Des Weiteren stellt sich die Frage, ob für die Vielzahl bestehender Gesetze und Verordnun-

gen rückwirkend Modelle beschlossen werden sollten. Da dies alleine schon aufgrund des Umfangs schlichtweg nicht durchführbar erscheint, ist es u.E. zweckmäßig, sich zunächst auf bestehende und nicht auf bereits abgeschlossene Gesetzgebungsverfahren zu konzentrieren. Da jedoch, wie beispielhaft für die steuerlichen Fristen in Abschnitt 3.3.1 gezeigt, in einem Gesetz auf Regelungen desselben Gesetzes oder auf Regelungen anderer Gesetze verwiesen werden kann, stellt sich die Frage, wie mit Regelungen, auf die verwiesen wird, verfahren werden soll, soweit sie nicht Teil von bestehenden Gesetzgebungsverfahren unter Einsatz von Modellen sind.

U.E. sollten losgelöst von den Grenzen der mittels DSL umzusetzenden, ausgewählten Gesetzgebungsverfahren sämtliche Regelungen in Modellen ausgedrückt werden, die auch mittelbar Einfluss auf die Regelungen haben, die durch das Gesetzgebungsverfahren eingeführt oder geändert werden. Für das Beispiel der Fristen aus Abschnitt 3.3.1 würde dies bedeuten, dass auch die entsprechenden Regelungen im BGB, auf die verwiesen wird, nachträglich durch Modelle beschrieben werden sollten. Durch ein solches Vorgehen können *in kleinen Schritten* bestehende Regelungen in das von uns vorgeschlagene System *überführt* werden.

5. ZUSAMMENFASSUNG UND AUSBLICK

Diese Studie hat gezeigt, dass man mittel- bis langfristig eine Unterstützung für die *Digitalisierung des Gesetzgebungsprozesses* benötigt, um die *Souveränität des Staates zu erhöhen*, also die Wirksamkeit, Adressatentauglichkeit und Vollzugsfähigkeit zu verbessern. Die Verwendung von DSLs und Nutzung von Modellen in den unterschiedlichen Schritten des Gesetzgebungsprozesses bietet *Vorteile für alle Nutzergruppen*: Vergleich von Modellen und Simulation in der Willensbildung, Simulation unterschiedlicher Modelle durch die Ministerialbürokratie, Vergleich von Lösungsalternativen und Feststellung der Digitaltauglichkeit in der Gesetzesvorbereitung in der ministeriellen Gesetzesvorbereitung und durch die Parlamentarier vor der Beschlussfassung, Rechtskonsistenzprüfungen durch neutrale Stellen, Analyse von Gesetzesänderungen durch Betroffene und Vergleich von geplanten mit realen Auswirkungen durch Parteien und Ministerialbeamte.

Mit der Verwendung von DSLs können *Effizienzsteigerungen* sowohl im Gesetzgebungsprozess als auch in der *Umsetzung der Gesetze* erreicht werden. Werden DSLs und Berechnungsmodelle als Standards und Normen definiert, die sowohl in der zuständigen Behörde, als auch in der unternehmensinternen Software direkt zur Ausführung kommen können, so führt dies zu einer effizienteren IT und besserer Zusammenarbeit der IT einzelner Verwaltungen und Unternehmen.

Die Studienautoren gehen davon aus, dass als Vorarbeit für die Erstellung von Modellen weiterhin ein aus Sicht der IT informeller juristischer Gesetzestext als Vorlage ausgearbeitet werden wird. Je früher im Rahmen des Gesetzgebungsverfahrens die jeweiligen Modelle mittels der standardisierten DSLs erarbeitet werden, desto weniger Übereinstimmungsprobleme sollten zwischen dem informell juristischen Text und den Modellen zu erwarten sein.

LITERATUR

Anzinger, H. M.: *Legal Tech in der Juristenausbildung. Friedrich-Naumann-Stiftung für die Freiheit*, 2020.

Barthel, C.: *Managementmoden in der Verwaltung: Sinn und Unsinn. Gabler Verlag*, 2020.

Beha, S.: *BFH-Präsident: „Es darf nicht passieren, dass Programmierer Herrschaft über das Gesetz übernehmen“*. Haufe Taxulting, 2019.

BMF Monatsbericht November 2019: 100 Jahre AO, URL: https://www.bundesfinanzministerium.de/Monatsberichte/2019/11/Inhalte/Kapitel-3-Analysen/3-6-100-jahre-abgabenordnung_pdf.pdf?__blob=publicationFile&v=3, 2019.

Borst, W.: *Construction of Engineering Ontologies. PhD thesis, Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands*, 1997.

Blumauer, A., Pellegrini, T.: *Semantic Web und semantische Technologien. Zentrale Begriffe und Unterscheidungen. In: Semantic Web. Wege zur vernetzten Wissensgesellschaft. Berlin: Springer Verlag*, S. 9-27, 2006.

Bench-Capon, T., Prakken, H.: *Introducing the Logic and Law Corner. Journal of Logic and Computation* 18(1), S. 1-12, 2007.

Deutscher Bundestag, *Datenhandbuch zur Geschichte des Deutschen Bundestages, Kapitel 10.1 Statistik zur Gesetzgebung*, URL: https://www.bundestag.de/resource/blob/196202/ee30d500ea94ebf8146d0ed7b12a8972/Kapitel_10_01_Statistik_zur_Gesetzgebung-data.pdf, 2018.

Drave, I., Henrich, T., Hölldobler, K., Kautz, O., Michael, J., Rumpe, B.: *Modellierung, Verifikation und Synthese von validen Planungszuständen für Fernsehstrahlungen. In: Modellierung 2020, Vienna*, S. 173-188, Gesellschaft für Informatik e.V., Feb. 2020.

Drüen, K.-D.: *100 Jahre Abgabenordnung, Aufsatz, DStR 2019*, S. 2657.

France, R., Rumpe, B.: *Model-driven Development of Complex Software: A Research Roadmap. Future of Software Engineering (FOSE '07)* S. 37-54, 2007.

Guarino N., Oberle D., Staab S.: *What Is an Ontology?. In: Staab S., Studer R. (eds) Handbook on Ontologies. International Handbooks on Information Systems. Springer*, 2009.

Gröpl, C.: *Staatsrecht I: Staatsgrundlagen, Staatsorganisation, Verfassungsprozess. 9. Auflage, C. H. Beck, München 2017*, Rn. 1114. Gruber, T. R.: *A Translation Approach to Portable Ontologies. Knowledge Acquisition*, 5(2), S. 199-220, 1993.

Hansen-Schirra, S., Neumann, S.: *Linguistische Verständlichmachung in der juristischen Realität. In: Lerch, Kent D. (Hg.): Die Sprache des Rechts ; Bd.1: Recht verstehen. Berlin: de Gruyter*, S. 167-184, 2004.

Hoekstra R., Breuker, J., Di Bello, M., Boer, A.: *The LKIF Core Ontology of Basic Legal Concepts. In Proc. 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques, volume 321. CEUR Workshop Proceedings, June 2007*.

Hodges, W.: *Formal Features of Compositionality*. *Journal of Logic, Language and Information* 10, 7-28, 2001.

Kar, R. M., Thapa, B. E. P., Hunt, S. S., Parycek, P.: *Recht Digital – Maschinenverständlich und automatisierbar. Impuls zur digitalen Vollzugstauglichkeit von Gesetzen*. Kompetenzzentrum Öffentliche IT, Fraunhofer FOKUS, 2019.

Lang, M.: *Unified Legal Language: Formalisierung der Methodik in der Rechtsanwendung und Entwicklung einer juristischen Metasprache – Dargestellt am Beispiel der Vermögensübertragung gegen wiederkehrende Leistungen*. Dissertation, Universität Augsburg, 2002.

Lusti, M.: *Expertensysteme im Recht*. *Internet-Zeitschrift für Rechtsinformatik und Informationsrecht*, S. 77-84, 1986.

McCarty, L. T.: *A Language for Legal Discourse: I. Basic Features*. In: *Proceedings of the 2nd International Conference on Artificial Intelligence and Law*, S. 180-189. ACM, 1989.

McCarty, L. T.: *Artificial Intelligence and Law: How to Get There from Here*. *Ratio Juris* 3(2), 189-200, 1990.

McCarty, L. T.: *Deep Semantic Interpretations of Legal Texts*. In: *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, S. 217-224. ACM, 2007

McKinsey & Company, Inc.: *Erst der Inhalt, dann die Paragraphen. Gesetze wirksam und praxistauglich gestalten*. Nationaler Normenkontrollrat, 2019

Meurers, W. D., Müller, S.: *Corpora and syntax*. In: *CorpusLinguistics. An International Handbook*. Teilband 2. Berlin, New York: Mouton de Gruyter, S. 920-933, 2009.

Mommers, L.: *Ontologies in the Legal Domain*. In: Poli R., Seibt J., editors, *Theory and Applications of Ontology: Philosophical Perspectives*, S. 265-276, Springer, 2010

Off, T., Kühn, H., Schuppan, T.: *Semantikbasierte und prozessorientierte E-Gesetzgebung zur Vollzugsoptimierung*. In: Rätz, D., Breidung, M., Lück-Schneider, D., Kaiser, S. & Schweighofer, E. (Hrsg.), *Digitale Transformation: Methoden, Kompetenzen und Technologien für die Verwaltung*. Bonn: Gesellschaft für Informatik e.V., S. 35-50, 2016.

Ruß, J., Ismer, R., Margolf, J.: *Digitalisierung des Steuerrechts: Eine Herausforderung für die Ausgestaltung von materiellen Steuergesetzen*, *DStR Heft 9*, S. 409-418, 2019.

Sherwin, E.: *Legal Taxonomy*. In *Legal Theory* 15; S. 25, 2009.

Sowizral, H. A., Kipps, J. R.: *ROSIE: A Programming Environment for Expert Systems*. Rand, 1985.

Schuppan, T., Köhl, S., Off, T.: *Vollzugsorientierte Gesetzgebung durch eine Vollzugssimulationsmaschine*. *Berichte des NEGZ*, Nr. 1, 2018.

Smith, B., Kusnierczyk, W., Schober, D., Ceusters, W.: *Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain*. *KR-MED 2006 "Biomedical Ontology in Action"*, 2006.

Soares, A. A., Martins P. V., da Silva, A. R.: *LegalLanguage: A Domain-Specific Language for Legal Contexts*. In David Aveiro, Giancarlo Guizzardi, and José Borbinha, editors, *Advances in Enterprise Engineering XIII*, pages 33-51. Springer, 2020.

Stachowiak, H.: *Allgemeine Modelltheorie*. Springer, 1973.

Tipke, K.: Steuergesetzgebung in der BR Deutschland aus der Sicht des Steuerrechtswissenschaftlers, StuW 1976, S. 293ff.

Thiemann, P.: Einführung in die denotationelle Semantik. In: Grundlagen der funktionalen Programmierung. Leitfäden der Informatik. Vieweg+Teubner Verlag, 1994.

Visser, P. R. S., Bench-Capon, T.J.M. A Comparison of Four Ontologies for the Design of Legal Knowledge Systems. Artificial Intelligence and Law 6, S. 27-57, 1998.

Völter, M., Benz, S., Dietrich, C., Engelmann, B., Helander, M., Kats, L. C. L., Visser, E., Wachsmuth, G.: DSL Engineering – Designing, Implementing and Using Domain-Specific Languages. dslbook.org, 2013.

Völter, M., Stahl, T., Bettin, J., Haase, A., Helsen, S., Czarnecki, K.: Model-Driven Software Development: Technology, Engineering, Management. Wiley Software Patterns Series, Wiley, 2013.

ÜBER DIE AUTORINNEN UND AUTOREN

Prof. Dr. Bernhard Rumpe leitet den Lehrstuhl Software Engineering der RWTH Aachen. Er beschäftigt sich mit domänenspezifischer Modellierung und ihrer Anwendung im Software und Systems Engineering, in der wissenschaftlichen Modellbildung oder auch der Vertragsgestaltung.

Mehrere dafür geeignete Sprachen, u.a. Derivate der UML und der SysML, wurden für den praktischen Einsatz auf Basis der Language Workbench MontiCore entwickelt. Prof. Rumpe ist unter anderem Mitglied im Center for Systems Engineering (CSE) und des Exzellenzclusters „Internet of Production“ der RWTH Aachen. Prof. Rumpe hat eine Reihe von Tagungen organisiert und ist Autor und Editor von 34 Büchern sowie Editor-in-Chief und Gründer des internationalen Journals on Software and Systems Modeling (SoSyM).

Dr. Judith Michael ist PostDoc und Teamleiterin für Modellbasierte Assistenz- und Informationsservices am Lehrstuhl für Software Engineering der RWTH Aachen sowie Aufsichtsratsmitglied Lakeside Science & Technology Park GmbH in Österreich. Ihre Forschung umfasst die (konzeptuelle) Modellierung von z.B. menschlichen Verhaltenszielen für Assistenzsysteme, Umgebungs- und Kontextinformationen und Datenschutzerklärungen. Zudem beschäftigt sie sich mit der modellbasierten Entwicklung von Assistenz- und Informationssystemen sowie der Konzeption der entsprechenden Softwarearchitekturen in den Anwendungsgebieten Ambient Assisted Living, Controlling und Finanzen, Energie, Gesundheit, Produktionssysteme, Internet of Things, Industrie 4.0 und Smart Homes.

Dr. Oliver Kautz ist PostDoc am Lehrstuhl für Software Engineering der RWTH Aachen. Seine Forschungsinteressen sind die Semantik von Modellierungssprachen, Evolutionsanalysen von Modellen, Software Language Engineering und die Entwicklung domänenspezifischer Sprachen, insbesondere natürlichsprachlicher Sprachen und der Weiterverarbeitung der Modelle dieser Sprachen zur Bereitstellung von automatisierten Werkzeugen zur Anwenderunterstützung.

Diplom-Finanzwirt (FH) Roland Krebs ist Referatsleiter und stellvertretender Abteilungsleiter beim Bayerischen Landesamt für Steuern (BayLfSt). Er leitet das Verfahren ELSTER (ELEktronische STEuerERklärung) seit dessen Einführung im Jahr 1999. ELSTER ist ein gemeinsames Projekt der Steuerverwaltungen der 16 Bundesländer und des Bundesministeriums der Finanzen zur sicheren elektronischen Übermittlung von Steuerdaten. ELSTER ist das bekannteste Verfahren im Rahmen von KONSENS (KoOrdinierte Neue Software-ENTwicklung der Steuerverwaltung). ELSTER ist zugleich das erste und größte E-Government Portal der Verwaltung. Die Projektleitung liegt beim Bayerischen Landesamt für Steuern.

Dipl.-Inf. Sabine Gandenberger ist Software-Architektin bei mgm technology partners gmbh in München. Als Projektleiterin der ELSTER Modellierungsplattform trägt sie maßgeblich dazu bei die Digitalisierung der Finanzverwaltung voranzutreiben.

Dipl.-Kfm. Janos Standt ist Bereichsleiter für den öffentlichen Sektor und Niederlassungsleiter der Standorte Köln und Aachen bei mgm technology partners gmbh. Gemeinsam mit Institutionen wie dem Bayerischen Landesamt für Steuern bringt mgm seit 20 Jahren Anwendungssysteme im öffentlichen Sektor in Produktion. Heute übermitteln mehr als 20 Millionen Bürger und mehr als 2 Millionen Unternehmen in Deutschland Daten über mgm-Software. Janos Standt und mgm legen

einen besonderen Fokus auf steuerliche Anwendungen sowie die Digitalisierung von Verwaltungsleistungen für Bürger und Organisationen. Aufgrund dieser Erfahrung und mithilfe der mgm-eigenen Entwicklungsplattform A12 ist mgm in einer idealen Ausgangsposition, um die zeitgerechte OZG-Umsetzung in Zusammenarbeit mit Ländern und Behörden zu ermöglichen.

StB Uli Weber, M.Sc. ist Tax Specialist/Syndikus-Steuerberater bei mgm technology partners gmbh in Köln. Er beschäftigt sich mit der Optimierung, Automatisierung und Digitalisierung von steuerlichen Prozessen und entwickelt modellbasierte, digitale Lösungen für den steuerlichen Bereich. Zudem vertritt er mgm technology partners gmbh im Institut für Digitalisierung im Steuerrecht e. V. (IDSt) und bringt seine Expertise im I. Fachausschuss „Digitalisierbarkeit von Steuernormen“ ein.

KOMMENTAR

„Im demokratischen Rechtsstaat regelt der Gesetzgeber, an welche Handlungen und wirtschaftlichen Sachverhalte er die Pflicht zur Steuerzahlung knüpft. Dabei sind Finanzverwaltung, Steuerpflichtige und deren Berater darauf angewiesen, dass sich die steuerlichen Rechte und Pflichten möglichst eindeutig, klar und präzise aus dem Gesetz ergeben. Die vorliegende NEGZ Studie entwickelt das Modell einer Domänen-spezifischen Sprache (Domain Specific Language, DSL), die nicht nur für alle Beteiligten eine größere Rechtssicherheit und Rechtsklarheit ermöglichen soll, sondern zugleich auch den Vorteil hätte, unmittelbar im digitalen Besteuerungsverfahren eingesetzt zu werden. Es handelt sich um ein wichtiges Projekt, das einen Fortschritt für die Digitalisierung im Steuerrecht bedeuten könnte. Die Digitalisierung der Gesetzgebung verdient weiter verfolgt zu werden. In diesem Zusammenhang werden dann auch die verfassungsrechtlichen Fragen zu würdigen sein, wie z.B. ob ein Gesetz in Form einer DSL verabschiedet werden kann.“

Prof. Dr. h.c. Rudolf Mellinghoff
Zentrum für Digitalisierung des Steuerrechts
der Ludwig-Maximilians-Universität München
(LMUDigiTax)

IMPRESSUM

Die Kurzstudie basiert auf einer Initiative des Nationalen E-Government Kompetenzzentrums e. V.

Ansprechpartner

Prof. Dr. Bernhard Rumpe

RWTH Aachen,
Lehrstuhl für Software Engineering
rumpe@se-rwth.de

Dr. Judith Michael

RWTH Aachen,
Lehrstuhl für Software Engineering
michael@se-rwth.de

Dr. Oliver Kautz

RWTH Aachen,
Lehrstuhl für Software Engineering
kautz@se-rwth.de

Roland Krebs

Bayerisches Landesamt für Steuern

Sabine Gandenberger

mgm technology partners gmbh
sabine.gandenberger@mgm-tp.com

Janos Standt

mgm technology partners gmbh
janos.standt@mgm-tp.com

Uli Weber

mgm technology partners gmbh
uli.weber@mgm-tp.com

Nationales E-Government Kompetenzzentrum e. V.

Pressehaus / 4102
Schiffbauerdamm 40
10117 Berlin

+49 (0)30 80494747
info@negz.org
negz.org

Gestalterische Umsetzung

made in – Design und Strategieberatung
www.madein.io

BERICHTE DES NEGZ

Folgende Kurzstudien sind in der Reihe „Berichte des NEGZ“ bereits erschienen:

- Nr. 1** Schuppan, T., Köhl, S., Off, T. (2018). Vollzugsorientierte Gesetzgebung durch eine Vollzugssimulationsmaschine, Berlin. » [DOI](#)
- Nr. 2** Ogonek, N., Distel B., Ben Rehouma, M., Hofmann, S., Räckers, M. (2018). Digitalisierungsverständnis von Führungskräften, Berlin. » [DOI](#)
- Nr. 3** Djeflal, C. (2018). Künstliche Intelligenz in der öffentlichen Verwaltung, Berlin. » [DOI](#)
- Nr. 4** Fadavian, B., Franzen-Paustenbach, D., Rehfeld, D., Schmitt, M., Schweikart, D., Djeflal, C. (2019). Data Driven Government, Berlin. » [DOI](#)
- Nr. 5** Balta, D., Hofmann, S., Rehfeld, D., Kuhn, P., Krcmar, H. (2019). Sharing Economy: Potential im öffentlichen Sektor, Berlin. » [DOI](#)
- Nr. 6** Hoepner, P., Welzel, C., Wulff, M. (2019). Identifizierung und Authentifizierung leicht gemacht – die Nutzer ins Zentrum stellen, Berlin. » [DOI](#)
- Nr. 7** Köhl, S., Müller, H. (2019). Sicherheitsanforderungen und -nachweise bei Cloud-Diensten – Grundlagen für öffentliche Auftraggeber, Berlin. » [DOI](#)
- Nr. 8** Houy, C., Gutermuth, O., Fettke, P., Loos, P. (2020). Potentiale Künstlicher Intelligenz zur Unterstützung von Sachbearbeitungsprozessen im Sozialwesen, Berlin. » [DOI](#)
- Nr. 9** Schaffer, S., Reithinger, N., Standt, J., Krebs, R. (2020). Sprachsteuerung von E-Government Diensten in Deutschland, Berlin. » [DOI](#)
- Nr. 10** Houy, C., Gutermuth, O., Fettke, P. (2020). Robotergestützte Prozessautomatisierung für die Digitale Verwaltung, Berlin. » [DOI](#)
- Nr. 11** Ogonek, N., Distel, B., Hofmann, S. (2020). Kompetenzvermittlung im öffentlichen Sektor neu gedacht, Berlin. » [DOI](#)
- Nr. 12** Distel, B., Hofmann, S., Østergaard Madsen, C. (2020). Nationale E-Government-Strategien: Deutschland und Dänemark im Vergleich, Berlin. » [DOI](#)
- Nr. 13** Halsbenning, S., Scholta, H., Distel, B. (2020). Quo vadis, Civis? Entwicklung einer Citizen Journey für eine nachfrageorientierte Dienstleistungsentwicklung im öffentlichen Sektor, Berlin. » [DOI](#)
- Nr. 14** Oschinsky, F., Stelter, A., Kaping, C., Niehaves, B. (2021). Kompetenzoffensive Bad Berleburg Digital (KOBOLD), Berlin. » [DOI](#)
- Nr. 15** Buchinger, M., Kuhn, P., Balta, D. (2021). Interoperabilität von Smart City-Datenplattformen, Berlin. » [DOI](#)

BERICHTE DES NEGZ

- Nr. 16** Löbel, S., Schuppan, T. (2021). Potenziale und Herausforderungen einer neuen Datenorientierung im Kontext öffentlicher Aufgabenwahrnehmung, Berlin. » [DOI](#)
- Nr. 17** Djeffal, C., Horst, A. (2021). Übersetzung und künstliche Intelligenz in der öffentlichen Verwaltung, Berlin. » [DOI](#)
- Nr. 18** Purgander, J., Hinckeldeyn, J. (2021). Untersuchung der Einsatzfähigkeit von Blockchain-Technologie für die Bauprüfung, Berlin. » [DOI](#)



**Nationales E-Government
Kompetenzzentrum e. V.**

Pressehaus/ 4102
Schiffbauerdamm 40
10117 Berlin

+49 (0)30 80494747
info@negz.org