

Stefan Leue, Rüdiger Wilhelmi

Modellierung des Rechts

Abstract: Modeling is a key ingredient of scientific reasoning in the technical and social sciences, and is also performed in law. In this paper we show that modeling methods from computer science, in particular mathematical logic and automated reasoning, can be successfully applied to the legal domain. While logical modeling (as well as automation of law) is only possible and practicable to a certain degree, we show that in the domain of contractual law that we consider it can be applied in a very useful manner. A case study illustrates that modeling and analysis methods adapted from automatic software verification can be successfully applied to the consistency analysis of legal contracts.

I Einführung

Die Modellierung von Sachverhalten und deren Nebenbedingungen ist eine grundlegende Aktivität in zahlreichen gesellschaftlichen Domänen. Modellierung spielt in der Wissenschaft, speziell in den Natur- und Ingenieurwissenschaften, eine zentrale Rolle. Modelle sind vereinfachende Abstraktionen der Realität. Sie dienen der Analyse, der Steuerung, der Simulation und dem Diskurs.

In diesem Artikel widmen wir uns der Frage, ob und wie juristische Sachverhalte mit Methoden der logischen Wissensmodellierung repräsentiert und analysiert werden können. Damit einher geht eine Diskussion der Grenzen der Modellierung juristischer Sachverhalte. Wir werden eine Modellierung vorstellen, die sich einem deutlich abgrenzbaren Sachgebiet, den Unternehmenskaufverträgen, widmet und gleichzeitig in der Analyse für wohldefinierte Fragestellungen, nämlich der internen Konsistenz des Vertragswerks, automatisch Antworten findet.

In unserer Arbeit lassen wir uns von Vorgehensweisen in dem Bereich des Software-Engineering leiten. Bei der Bestimmung von Kalkülen, mit denen sich Konsistenzfragen für Softwaresysteme modellieren und analysieren lassen, wird häufig auf die Beobachtung „software is logic at work“ Bezug genommen und die formale, mathematisch basierte Logik als ein geeigneter Kalkül vorgeschlagen. Diese Idee übertragen wir in unserer Arbeit auf die Modellierung und Analyse von Unternehmenskaufverträgen. Dabei zeigen wir auch Parallelen zwischen der objektorientierten Modellierung in der Informatik und der Systembildung im Recht auf.

Mit unserer Arbeit möchten wir schließlich einen Beitrag zu der kontroversen Diskussion um die Beziehung von Recht und logischer Modellierung leisten. Schon im 17. Jahrhundert schrieb *Leibniz* seine Bachelorarbeit zu dem Thema *Specimen*

certitudinis seu demonstrationum in jure exhibitum in doctrina conditionum, was sich übersetzen lässt als *Probe der Gewissheit oder Beweise im Recht ausgeführt in der Bedingungslehre*¹. Deutlich kritischer äußert sich *Holmes* im späten 19. Jahrhundert: „the life of the law has not been logic: it has been experience“.² Wir möchten mit dieser Arbeit demonstrieren, dass sich für wohl definierte, eingeschränkte Rechtsbereiche durch die gemeinsame Verwendung von mathematischer Logik und modernen algorithmischen Analyseverfahren wichtige und hilfreiche Einsichten in die logische Struktur von Verträgen – in unserem Fall Unternehmenskaufverträgen – ergeben, die während der Vertragsentwicklung von großer praktischer Hilfe sein können.

II Was ist Modellierung?

„All models are wrong, but some are useful“³

Der allgemeine Modellbegriff zeichnet sich durch drei Hauptmerkmale aus, das Abbildungsmerkmal, das Verkürzungsmerkmal und das pragmatische Merkmal:⁴ Modelle sind Abbildungen oder Repräsentationen von Originalen, welche natürlich oder künstlich sein können. Sie erfassen nicht alle Attribute des Originals, sondern sind auf diejenigen verkürzt, die den Modellierern relevant erscheinen. Insofern kann man davon sprechen, dass sie notwendig falsch sind. Modelle sind in dem Sinne pragmatisch, dass sie eine Funktion haben, indem sie auf bestimmte menschliche oder künstliche Modellnutzer zielen, auf bestimmte Zeiträume bezogen sind und bestimmten Zwecken im Rahmen gedanklicher oder tatsächlicher Operationen dienen. Wenn sie diese Funktion erfüllen, sind sie nützlich.

Konkret sind Modelle Abstraktionen der beobachteten oder postulierten Realität, die auf eine relevante Menge von Domänen-Merkmalen (in den Datenwissenschaften auch „Features“ genannt) fokussieren und irrelevante Merkmale vernachlässigen. Modelle sind Gegenstand der Analyse und des Diskurses sowie der Simulation und Steuerung. Sie dienen dementsprechend der Generierung von Wissen und Artefakten. Die mit der Modellierung einhergehenden Abstraktionen

1 Näher dazu *Armgardt*, Das rechtslogische System der „Doctrina conditionum“ von Gottfried Wilhelm Leibniz, 2001, S. 11 und *passim*.

2 *Holmes*, The Common Law, 1881, S. 5.

3 *Box*, Robustness in the strategy of scientific model building, in: Launer/Wilkinson (Hrsg.), Robustness in Statistics, 1979, S. 201, 202 f.

4 *Stachowiak*, Allgemeine Modelltheorie, 1973, S. 131 ff.; vgl. auch *Sorge*, Modellierung in Recht und Informatik, in: Oostrom/Weth (Hrsg.), Festschrift für Herberger, 2016, S. 877, 878.

erlauben eine substanzielle Komplexitätsreduktion, die bei der Betrachtung besonders komplexer Systeme deren Analyse überhaupt erst ermöglicht.

Modellierung ist damit stets eine Vereinfachung im Sinne einer Reduktion von Komplexität. Sie ist allerdings selbst in aller Regel alles andere als einfach, sondern setzt eine erhebliche Strukturierungsleistung voraus. Diese liegt etwa in der Herausarbeitung von Gemeinsamkeiten und Unterschieden, Zusammenhängen und Abhängigkeiten sowie Vererbungen und Verfeinerungen. Damit verbunden ist eine Übersetzung, die sich innerhalb natürlicher Sprachen auf Begriffe beschränken kann, aber häufig auch natürliche Sprache oder quantitative Daten in formale Sprachen, also mathematische oder logische Formalismen, oder Grafiken überträgt.

Wir betrachten in dieser Arbeit zwei spezielle Formen der Modellierung. Zunächst sind wir an ontologischer Modellierung der hier betrachteten Rechtsdomäne der Unternehmenskaufverträge interessiert. Im Mittelpunkt steht die Frage, welche zentralen Konzepte für die spätere logische Analyse von zentraler Bedeutung sind.

Dabei ist die Beschreibung von Beziehungsstrukturen zwischen Konzepten eine wichtige Komponente der ontologischen Modellierung. So möchte man gerne ausdrücken, dass ein Konzept aus mehreren Teilkonzepten besteht („ist Teil von“), oder dass ein Konzept eine Verallgemeinerung eines anderen ist, wie z. B. in der Beziehung zwischen den Konzepten Person, Käufer und Verkäufer ersichtlich. Dabei werden die Merkmale des allgemeineren Konzepts, z. B. der Name, von den spezialisierten oder verfeinerten Konzepten ererbt. Bei dieser Vererbung gibt es aber auch die Idee der Überschreibung (Polymorphie), bei der ein ererbtes Merkmal durch eine neue Charakterisierung unter Verwendung des Namens des allgemeinen Konzepts neu definiert wird. Dies hat besondere Bedeutung in der Rechtsmodellierung. So kann man durch Vererbung und Überschreibung zum Ausdruck bringen, dass alle mit Ausnahme der überschriebenen Regelungen bei der Vererbung einer allgemeinen Rechtsregel auf eine spezielle übernommen werden. Ein Beispiel ist das Verhältnis zwischen allgemeinen und besonderen Teilen im Recht, wo die Regelungen des allgemeinen Teils auch im besonderen Teil gelten, aber ergänzt oder verdrängt werden, soweit sich dort weitere oder abweichende Regelungen finden.

Ein weiterer Aspekt der Modellierung ist, dass wir die beschriebenen Beziehungsstrukturen von Konzepten sowohl zwischen Instanzen (Objekten), wie auch zwischen den Typen der Instanzen (Klassen) in unserer Modellwelt repräsentieren können. Weiterhin sind wir an der qualitativen Modellierung von Beziehungen zwischen Objekten bzw. Klassen untereinander interessiert, so z. B. der Tatsache, dass ein Eigentumsrecht das Eigentum zwischen einer Person und einem Objekt etabliert. Dieser Modellierungsansatz entspricht dem der objektorientierten Modellierung, die u. a. in dem Bereich der System- und Softwareentwicklung wohl etabliert ist.

Bei der weiteren Modellierung steht die Beschreibung von logischen Abhängigkeiten zwischen den in der ontologischen Modellierung definierten Objekten bzw.

Klassen im Mittelpunkt. So möchten wir beispielsweise sicherstellen, dass eine Eigentumsübertragung eines Kaufgegenstands vom Verkäufer auf den Käufer nur dann möglich ist, wenn der Verkäufer Eigentümer des Kaufgegenstands ist. Sollte es in dem Vertrag hinterlegte Fakten geben, dass z.B. das Eigentumsrecht an einem Kaufgegenstand verpfändet ist, so steht dies im Widerspruch zu der oben genannten Eigentumsbedingung, wodurch die Erfüllung des primären Anspruchs der Eigentumsübertragung unmöglich werden kann. Ziel dieser Modellierung ist die automatische Erkennung solcher Widersprüche, ein aus der Software-Analyse wohlbekanntes Verfahren.

III Warum (Recht) modellieren?

Nicht nur Informatiker, sondern auch Juristen sind als Modellierer tätig.⁵ In der Rechtswissenschaft werden Modellierungen traditionell im Bereich der Didaktik verwendet, um das Recht, insbesondere Rechtsinstitute und Rechtsbeziehungen zu visualisieren. Dabei kann man zwischen ikonischen und logischen Bildern unterscheiden.⁶ Ikonische Bilder ähneln als Zeichen dem Bezeichneten, man spricht auch von analogen oder mimetischen Bildern oder Abbildern. Das Spektrum reicht von naturalistischen und realistischen Bildern, die ein konkretes Modell wiedergeben, bis zu abstrahierenden und stilisierten Bildern, die nicht mehr auf Individuen, sondern nur noch auf Gattungen verweisen. Logische Bilder deuten formale Beziehungen an und visualisieren diese etwa durch Verbindungs- und Trennlinien, Begriffs- oder Entscheidungsbäume, Flussdiagramme und/oder Zeitstrahlen. Sie zeigen eine Struktur ihres Gegenstands auf und werden daher auch Strukturdiagramme genannt. Als solche spielen sie im juristischen Unterricht seit jeher eine große Rolle. In Bezug auf konkrete Anspruchsbeziehungen spricht *Heck* von juristischen Zeichnungen.⁷

Logische Bilder stellen eine Modellierung dar oder beruhen jedenfalls auf einer solchen. Ihre Verwendung setzt eine entsprechende Analyse der zu modellierenden Struktur voraus. Diese ist nicht nur Voraussetzung für die Verwendung logischer Bilder in der Didaktik, sondern kann auch bereits der eigentliche Zweck der Modellierung sein, die dann Mittel der Analyse ist.

⁵ *Sorge*, Festschrift für Herberger (Fn. 4), S. 877, 878.

⁶ Vgl. *Ulbrich*, Didaktik visueller Rechtskommunikation, in: Krüper (Hrsg.), Rechtswissenschaft lehren, 2022, § 36, S. 896, 906 ff.

⁷ *Heck*, Grundriß des Schuldrechts, 1929, Vorwort.

Dabei kann bereits der Gesetzgeber als Modellierer betrachtet werden, da er Sachverhalte kategorisiert und daran Konsequenzen knüpft.⁸ Dementsprechend wird zum Teil auch Modellierungssoftware bei der Erarbeitung von Gesetzentwürfen verwendet.⁹ Der Gesetzgeber erklärt dabei bestimmte Attribute sozialer Beziehungen für rechtlich relevant und andere nicht. Dies ist nicht auf die Verknüpfung von Tatbeständen und Rechtsfolgen beschränkt. Indem Gesetze eine Vielzahl unterschiedlicher Normen und Normtypen aufweisen, die aufeinander bezogen sind und bestimmte Lebenssachverhalte auf der Grundlage von Wertungen zu bestimmten Zwecken regeln wollen, schaffen sie nicht nur Rechtsinstitute, unter denen ein „Normeninbegriff zur Regelung der Gerechtigkeits- und Ordnungsziele bei bestimmten sozialen Interessenkonstellationen“¹⁰ verstanden werden kann. Vielmehr weisen sie auch ein äußeres System im Sinne einer deskriptiven Ordnung des Rechts nach formalen Gesichtspunkten und ein inneres System im Sinne der inhaltlichen Widerspruchsfreiheit und Folgerichtigkeit der dem Recht zugrundeliegenden Wertungen auf.¹¹ Diese sind zum Teil als solches intendiert, erschließen sich aber häufig auch erst durch eine umfassende Analyse.

Diese Analyse ist Gegenstand der Rechtsdogmatik. Diese ordnet und systematisiert den Rechtsstoff und arbeitet das innere Wertsystem und generell anwendbare Lösungsmuster heraus, die die Rechtsanwendung stabilisieren und entlasten, aber auch die Grundlagen für die Weiterentwicklung des Rechts bilden.¹² Sie stellt nicht nur auf die Gesetze ab, sondern bezieht auch die Rechtsprechung und die Verwaltungs- und Vertragspraxis sowie die Literatur mit ein. Diese Funktion der Rechtsdogmatik ist im kontinentaleuropäischen Rechtskreis mit seinen Kodifikationen besonders ausgeprägt, gilt aber etwa auch für den anglo-amerikanischen Rechtskreis, auch wenn dort ein weniger hoher Abstraktionsgrad angestrebt wird.

Die Analyse umfasst dabei jedenfalls im theoretischen Ausgangspunkt alle geregelten Fallkonstellationen, also nicht nur die bereits entschiedenen, sondern auch die möglichen Fälle. Da diese jedoch nicht alle einzeln betrachtet werden können, bedarf es der Modellbildung, die häufig in natürlicher Sprache erfolgt, sich aber

⁸ *Sorge*, Festschrift für Herberger (Fn. 4), S. 877, 878.

⁹ Vgl. *Breidenbach/Schmid*, in: Breidenbach/Glatz (Hrsg.), *Rechtshandbuch Legal Tech*, 2021, S. 255 ff.

¹⁰ *Westermann*, Vertragsfreiheit und Typengesetzlichkeit im Recht der Personengesellschaften, 1970, S. 58 m. Fn. 7.

¹¹ Näher zu der Unterscheidung *Rüthers/Fischer/Birk*, *Rechtstheorie*, 10. Aufl. 2018, Rn. 139 ff.; *Röhl/Röhl*, *Allgemeine Rechtslehre*, 3. Aufl. 2008, S. 438 ff.

¹² Vgl. *Rüthers/Fischer/Birk*, *Rechtstheorie*, 10. Aufl. 2018, Rn. 321 m. Verw. auf *Luhmann*, *Rechtssystem und Rechtsdogmatik*, 1974, S. 19 ff.

auch grafischer Darstellungen oder Formalisierungen bedienen kann. Rechtswissenschaft ist insofern – zumindest auch – Strukturwissenschaft.¹³

Die Modellierung und insbesondere Formalisierung können nicht nur zur Analyse erfolgen, sondern auch mit dem Ziel der maschinellen oder automatischen Verarbeitung. Eine erste Stufe bildet hier die Entwicklung juristischer Datenbanken, insbesondere von Juris, mit der Formalisierung von Rechtsdokumenten zum Zwecke des Information Retrieval. Rechtsanwendung im weiteren Sinne stellt demgegenüber ein Tool wie *flightright.de* dar, das stochastische Methoden mit sehr engen und schematischen Anspruchsvoraussetzungen und einer Vielzahl gleich- oder sehr ähnlich gelagerter Fälle kombiniert¹⁴ und daher einen Sonderfall darstellt. Auch hier setzt die Programmierung jedoch eine entsprechende Modellierung voraus.

Unser im folgenden behandeltes Beispiel der Modellierung dient demgegenüber nicht der Vorhersage oder Ersetzung von Rechtsanwendung und insbesondere Rechtsprechung,¹⁵ sondern möchte die Rechtsanwendung lediglich unterstützen. Die Modellierung von Verträgen dient hier der Analyse der Konsistenz der vertraglichen Regelungen, indem sie es ermöglicht, dass entsprechende Programme konkrete Verträge prüfen und auf problematische Regelungen hinweisen.

IV Was im Recht modellieren?

In den bisherigen Ausführungen sind wir davon ausgegangen, dass das Recht prinzipiell modelliert werden kann, ohne näher zu betrachten, was im Recht modelliert werden soll. Dies hängt vom Ziel der Modellierung ab.

So kann man die Modellierung auf die Gesetze beschränken. Dies kann genügen, wenn es um die Struktur von gesetzlichen Normen geht. Allerdings wird damit

13 Vgl. *Hufen*, Der wissenschaftliche Anspruch des Jurastudiums, *JuS* 2017, 1, 5; grdlg. *Müller*, Strukturierende Rechtslehre, 2. Aufl. 1994; engerer – an Mathematik und Informatik angelehnter – Begriff bei *Steininger*, Die Jurisprudenz auf Erkenntnissuche? – Ein Plädoyer für eine Neuorientierung der Rechtswissenschaft, *NJW* 2015, 1072, 1074 ff.

14 Vgl. *Broemel*, Algorithmen in der Rechtsanwendung, in: *Kuhli/Rostalski* (Hrsg.), Normentheorie im digitalen Zeitalter, 2023, S. 123, 127 ff.; näher zu *flightright.de* *Timmermann*, Legal Tech-Anwendungen, 2020, S. 170 ff.

15 Vgl. zur Subsumtion durch Algorithmen *Timmermann/Gelbrich*, Können Algorithmen subsumieren? Möglichkeiten und Grenzen von Legal Tech, *NJW* 2022, 25 ff., *passim*. Vgl. auch die Kontroverse zwischen *Kotsoglu*, Subsumtionsautomat 2.0: Über die (Un-)Möglichkeit einer Algorithmisierung der Rechtserzeugung, *JZ* 2014, 451; *Engel*, Erwiderung: Algorithmisierte Rechtsfindung als juristische Arbeitshilfe, *JZ* 2014, 1096; *Kotsoglu*, Schlusswort – „Subsumtionsautomat 2.0 reloaded?“ – Zur Unmöglichkeit der Rechtsprüfung durch Laien, *JZ* 2014, 1100 zu *Raabe* u. a., Recht ex machina, 2012.

vor allem die äußere Struktur abgebildet. Nicht erfasst werden implizite, insbesondere vorausgesetzte Informationen und Konkretisierungen. Diese sind in der Regel nicht nur zur Anwendung der Normen erforderlich, sondern auch um das innere System zu erfassen, das den inhaltlichen Zusammenhang zwischen Regelungen und die zugrundeliegenden Wertungen umfasst.

Will man hingegen das Recht im Sinne der geltenden bzw. Anwendung findenden rechtlichen Regelungen erfassen, müssen nicht nur die gesetzlichen Normen modelliert werden, sondern auch das Gewohnheitsrecht und vor allem die Rechtsprechung und die Verwaltungs- und Vertragspraxis sowie die juristische Literatur. Denn auch diese sind bei der Lösung von Rechtsproblemen zu berücksichtigen und auch auf ihnen beruht die Rechtsdogmatik, die das Recht für konkrete Fälle konkretisiert oder zumindest die Grundsätze und Argumentationsfiguren bereitstellt, die eine weitere Konkretisierung erlauben.

In der Rechtspraxis empfiehlt sich bei der Modellierung die Orientierung an den Rechten und Ansprüchen, die zu prüfen sind. Geht es um die Herausarbeitung der zugrundeliegenden Wertungen, kann es sich aber auch empfehlen, von Rechtsinstituten auszugehen.

Eine solche Modellierung ist aber – auch wenn sie nur auf Teilbereiche des Rechts bezogen ist – sehr aufwändig. Sie muss nicht nur klären, was gilt oder zumindest als geltend unterstellt wird, was angesichts der Meinungsfreude der Juristen – drei Juristen fünf Meinungen – und häufig divergierender insbesondere untergerichtlicher Rechtsprechung schwierig genug ist. Vor allem aber steht sie vor der Herausforderung, auch die noch nicht entschiedenen Fälle und Fallkonstellationen zu erfassen. Das bereits erwähnte Tool flightright.de stellt insofern eine seltene Ausnahme dar. Ähnlich wie man die Begriffsjurisprudenz als Rahmen auffassen kann, der insbesondere bei der Rechtsanwendung mit Hilfe der Wertungsjurisprudenz auszufüllen ist, kann die Modellierung das Recht nur bis zu einem gewissen Grad abbilden. Auf absehbare Zeit realistischer ist es daher, wenn die Modellierung die Juristen bei ihrer Tätigkeit unterstützt, indem sie das relevante Recht vorstrukturiert und auf möglicherweise problematische Stellen hinweist, die dann der näheren händischen Prüfung bedürfen – etwa durch Warnungen in einem „Red Flag“-System (siehe unten VIII.4.).

Modelliert werden können auch Verträge, um diese zu analysieren oder auch zu generieren. Darunter könnte man sog. Smart Contracts zählen, die aber nach einem gängigen Bonmot weder smart noch Contract sind, aber die rechtlichen Verpflichtungen aus dem Vertrag, also die danach notwendigen Aktionen mit ihren tatsächlichen Voraussetzungen modellieren. Sie werden hier jedoch nicht vertieft. Möglich ist aber auch die Modellierung allgemeiner Verträge, um sie auf interne, logische Konsistenz und Vollständigkeit zu analysieren. Das ist der Ansatz unseres Beispiels.

V Was für Methoden der Modellierung gibt es?

Die Natur der Modelle in unterschiedlichen Domänen ist vielfältig. Beispiele für Modelle sind Klima- und Wettermodelle, bei denen kurz- oder langfristige Entwicklungen durch Simulation prognostiziert werden, Architekturmodelle, die durch Werkzeuge der virtuellen Realität eine Vorabbegutachtung von Bauvorhaben erlauben, oder große Sprachmodelle, die dem Verständnis und der Generierung von natürlicher Sprache dienen.

1 Logische Modellierung

Wir wollen uns in dieser Arbeit zur Repräsentation des logischen Gehalts eines Vertrags einer Modellierungstechnik bedienen, die auf der Modelltheorie im Sinne von *Tarski* beruht.¹⁶ Die grundlegende Idee ist dabei, das Modell als eine Folge von Zuweisung von Wahrheits- oder Domänenwerten an Variablen bzw. Prädikate zu verstehen. Eine solche Zuweisung, die auch als „Welt“ bezeichnet wird, ist damit so etwas wie ein Schnappschuss der Realität. Um ein Beispiel zu betrachten, nehmen wir an, dass Eva ihre Bäckerei für einen Kaufpreis von EUR 100.000 an Chris verkaufen möchte und zu diesem Zweck einen Vertrag mit Chris geschlossen hat. Wir können jetzt die Tatsache, dass Eva Eigentümerin der Bäckerei ist, dadurch modellieren, dass die logischen Prädikate *Eigentum(Bäckerei) = Eva* und *Eigentum(Kaufpreis) = Chris* wahr sind. Wenn der Vertrag vollzogen ist, dann gilt *Eigentum(Bäckerei) = Chris* und *Eigentum(Kaufpreis) = Eva*. Sollten wir jedoch durch die Ausführung eines Vertragsschritts in eine Welt geraten, in der *Eigentum(Bäckerei) = Chris* und *Eigentum(Kaufpreis) = Chris* gelten, dann ist dies Evidenz für ein Konsistenzproblem innerhalb des Vertrages, da offenkundig eine Eigentumsübertragung ohne Zahlung des Kaufpreises stattgefunden hat.

Für die Modellierung der logischen Struktur von Gesetzen oder Verträgen sind in der Vergangenheit zahlreiche Modellierungssprachen entwickelt worden. Wir wollen an dieser Stelle nur einige dieser Sprachen beispielhaft erwähnen. Die Sprache *Symboleo*¹⁷ dient ähnlich wie unser unten beschriebener Ansatz der Modellierung von Verträgen. Allerdings ist sie nicht auf eine formale Analyse ausge-

¹⁶ *Tarski*, Contributions to the theory of models, *Indagationes Mathematicae* 57 (1954), S. 572 ff.

¹⁷ *Sharifi/Parvizimosaed/Amyot/Logrippo/Mylopoulos*, *Symboleo: Towards a Specification Language for Legal Contracts*, in: 2020 IEEE 28th International Requirements Engineering Conference (RE), 2020, S. 364 ff.

richtet. eFlint¹⁸ ist eine domänenspezifische Sprache zur Modellierung von Rechtsnormen. Sie bietet Möglichkeiten zur automatischen Fallbewertung und manuellen Ausführung. Die von Microsoft entwickelte Sprache Catala¹⁹ erlaubt die direkte Modellierung von Gesetzesrecht in der Form von ausführbaren Programmen, und könnte damit als Programmiersprache für Gesetze bezeichnet werden.

Bei der Wahl einer geeigneten logischen Modellierungsmethode beziehen wir uns auf Erfahrungen aus dem Software-Engineering und insbesondere der Softwareverifikation. Dort gibt es zahlreiche sehr erfolgreiche Methoden, die unter anderem auf den Ideen der Modelltheorie beruhen. Zu diesen gehören das interaktive Theorembeweisen²⁰, die Modellprüfverfahren (Model Checking)²¹ und die Erfüllbarkeitsüberprüfung Modulo Theorien (*Satisfiability Modulo Theories*, SMT)²², die alle zu sehr erfolgreichen Anwendungen in der Verifikation von Hard- und Softwaresystemen geführt haben. Fokussieren wollen wir uns dabei auf die Verwendung von SMT-basierten Modellierungs- und Analyseansätzen. Der Grund ist der, dass diese auf Fragmenten der Logik erster Stufe basieren, für die das Erfüllbarkeitsproblem entscheidbar ist. Eine logische Formel ist erfüllbar, wenn es eine Belegung der in ihr enthaltenen logischen Variablen (Propositionen) gibt, so dass die Formel wahr ist. Im Sinne der Modelltheorie bedeutet dies, dass die Formel ein erfüllendes Modell hat. Das Problem der Erfüllbarkeit einer logischen Formel ist entscheidbar, wenn es einen terminierenden Algorithmus gibt, der die Frage der Erfüllbarkeit einer beliebigen solchen Formel mit wahr oder falsch beantwortet. Bei der Verwendung von beliebiger Logik erster oder höherer Stufe, wie z. B. beim interaktiven Theorembeweisen, oder nicht-monotonen Logiken, ist diese Entscheidbarkeit nicht gegeben. Wir verwenden daher in unserer logischen Modellierung lediglich Quantoren-freie entscheidbare Fragmente der Logik erster Stufe, für die es effiziente, in SMT-Solvern implementierte Entscheidungsverfahren gibt. Die Verwendung der SMT-Technologie erlaubt also die Entwicklung vollständig automatisierbarer Analyseverfahren, die keine Intervention durch den Benutzer verlangen und sich damit, basierend auf den Erfahrungen mit der Verwendung automatischer formaler Verifikationsmethoden im Software Engineering, sehr gut in Vertrags-

18 van Binsbergen/Liu/van Doesburg/van Engers, EFLINT: a domain-specific language for executable norm specifications, in: Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences (GPCE 2020), 2020, S. 124 ff.

19 Merigoux/Chatain/Protzenko, Catala: a programming language for the law, in: Proceedings of the ACM on Programming Languages, 5, ICFP, 2021, S. 1 ff.

20 Nipkow/Paulson/Wenzel, A Proof Assistant for Higher-Order Logic. Lecture Notes in Computer Science, 2002.

21 Clarke/Henzinger/Veith/Bloem, Handbook of Model Checking, 2018.

22 Barrett/Tinelli, Satisfiability Modulo Theories, in: Clarke/Henzinger/Veith/Bloem (Fn. 21), S. 305 ff.

verhandlungsprozesse integrieren lassen. Viele große Software-Entwicklungsfir-
men (z. B. Microsoft, Amazon, Google, Facebook) verwenden SMT-basierte Verfah-
ren in der Software-Qualitätssicherung.

2 Semantik der logischen Modellierung

Neben der Abstraktion ist eine mathematische Fundierung ein Kennzeichen der in
den Natur- und in den Ingenieurwissenschaften verwendeten Modelle. Die ma-
thematische Fundierung, auch als Semantik bezeichnet, erlaubt eine eindeutige
Interpretation der Modelle und eine Analyse oder Simulation durch Software-
werkzeuge. Meteorologische Modelle und Klimamodelle beruhen auf einer Kom-
bination von Differentialgleichungen und wahrscheinlichkeitstheoretischen Mo-
dellelementen. Große Sprachmodelle werden als statistische Modelle über
Worteinbettungen durch maschinelles Lernen aus großen Sprachkorpora erzeugt.

Softwaremodelle zeichnen sich dadurch aus, dass sie die diskrete Natur der
Ausführung von Programmen berücksichtigen. Dazu verwendet man häufig Mo-
dellierungsmethoden der diskreten Mathematik, speziell Automaten und formale
Logik. Wir betrachten in dieser Arbeit die Ausführung von Verträgen als diskrete
Sequenzen von Ausführungsschritten, wodurch sich die Verwendung dieser dis-
kreten, ursprünglich für Softwaremodelle vorgeschlagenen Beschreibungsmetho-
den anbietet. Mit einer Automatenmodellierung lässt sich z. B. beschreiben, dass das
Eigentum an der Bäckerei durch die Eigentumsübertragung von Eva auf Chris
übergeht. Mit formaler Logik lässt sich ausdrücken, dass die Eigentumsübertragung
nur dann ohne Weiteres erfolgen kann, wenn Eva als Verkäuferin auch Eigentü-
merin der Bäckerei ist, und diese nicht z. B. als Sicherheit an eine Bank übereignet
ist. Sonst müssten gegebenenfalls andere Vorkehrungen getroffen werden.

3 Ontologische Modellierung

Neben der Modellierung des logischen Gehalts eines Unternehmenskaufvertrags
widmen wir uns auch der Fragestellung, was ein Kaufvertrag ist, und welche
Schritte in der Ausführung des Kaufvertrags dessen Zustand bzw. den Zustand der
Welt verändern können. Es handelt sich hier also um eine ontologische Modellie-
rung. Wir verwenden hierzu die *Unified Modeling Language* (UML)²³ als Modellie-

²³ Object Management Group, OMG® Unified Modeling Language® (OMG UML®), Version 2.5.1,
2017, abrufbar unter <https://www.omg.org/spec/UML/2.5.1/PDF> (4.2.2024).

rungsnotation. Die UML unterstützt die oben diskutierten Modellierungsformen der Ist-Teil-Von Modellierung, der Verallgemeinerung, der Verfeinerung und des Überschreibens. Sie hat ihre Wurzeln in der Modellierung von Datenbanken und Softwarearchitekturen, wird aber auch in vielen weiteren Domänen zu Modellierung und Meta-Modellierung eingesetzt. Ihre Verwendung wird etwa auch für die Rechtsvisualisierung vorgeschlagen.²⁴

Die UML ist als graphische Modellierungssprache konzipiert. Unterschiedliche Typen von Diagrammen erlauben eine abstrakte Sichtweise auf unterschiedliche Aspekte eines UML-Modells. Jede Diagrammart verfügt über ein Meta-Modell, das die Struktur der Diagrammart charakterisiert. Zentrale Bedeutung haben die Objekt- und Klassendiagramme. Diese erlauben die Modellierung von Objekten einschließlich ihrer Attribute (Eigenschaften) und Operationen (Fähigkeiten). Ein Beispiel für die Klassenmodellierung eines Vertrages wird unten bei VIII.2. diskutiert.

Für eine ontologische Sichtweise von Systemen, und damit auch von soziotechnischen Systemen, ist die UML durch ihre Eignung für objektorientierte Modellierung hervorragend geeignet. Die Objektorientierung bedeutet, dass die Modellelemente realweltliche Objekte einschließlich derer Charakteristika (Attribute) und Fähigkeiten (Operationen) direkt mit den Elementen des Modells korrespondieren. Einzelne Objekte mit identischen Attributen und Operationen können zu Klassen aggregiert werden, wodurch die ontologische Modellierung unterstützt wird. So definieren wir ein Klassendiagramm zur Beschreibung der Ontologie von Unternehmenskaufverträgen, und können dann eine konkrete Vertragsinstanz als ein mit dem Klassendiagramm kompatibles Objektdiagramm repräsentieren.

Um die Möglichkeiten der Klassenmodellierung und der Vererbung mittels der UML zu illustrieren, verweisen wir auf die Modellierung eines Unternehmenskaufvertrags in Abb. 2 in VIII.2. So können wir z.B. beschreiben, dass ein Unternehmenskaufvertrag aus einer unbeschränkten Anzahl primärer und sekundärer Ansprüche besteht. Gleichzeitig können wir ausdrücken, dass man primäre und sekundäre Ansprüche zu allgemeinen Ausdrücken verallgemeinern kann. Dies bedeutet andererseits gleichzeitig, dass primäre und sekundäre Ansprüche Verfeinerungen des allgemeinen Anspruchs darstellen und alle Eigenschaften des allgemeinen Anspruchs ererben.

Neben der ontologischen Modellierung der logischen Struktur von Verträgen ist die zeitliche Dynamik der Ausführung eines Vertrags von Interesse. Wie oben erläutert kann die Ausführung eines Vertrags durch Folgen von Zustandsübergängen erzielt werden. Im Rahmen der UML ist die Verwendung von Zustandsübergängen

²⁴ Kahlig/Kahlig, Rechtsvisualisierung – Viribus Unitis – mit C.O.N.T.E.N.T., Jusletter IT 26. Februar 2015.

in Form von Zustandsmaschinendiagrammen die geeignete Form der Modellierung zeitlicher Dynamik.

VI Herausforderungen der Modellierung?

1 Informationsverlust durch Abstraktion

Eine intrinsische Herausforderung der Modellierung ist der Informationsverlust durch Abstraktion, der zu einem Verlust an Korrespondenz der abstrakten Modellwelt und der konkreten Realwelt führen kann. Dies kann dazu führen, dass von einem Modell abgeleitete Informationen nicht mehr eindeutig in die Realwelt zurückübersetzt werden können. In der Softwareentwicklung begegnet man diesem Phänomen durch die Verwendung objektorientierter Modellierungskonzepte. Die zentrale Idee ist dabei die Korrespondenz von Objekten bzw. Klassen mit Objekten bzw. Klassen in der Softwarearchitektur. Bei der Modellierung von juristischen Artefakten ist eine objektorientierte Modellierung ebenfalls von großem Vorteil. Wie oben ausgeführt, liegt sie auch unserer Modellierung von Unternehmenskaufverträgen zugrunde, so etwa bei der Modellierung von Vertragsansprüchen.

2 Komplexität des Gegenstands Recht

Die Komplexität von technischen oder sozio-technischen Systemen ist eine weitere, wichtige Herausforderung für die Modellierung. Komplexität ergibt sich u. a. aus einer hohen Anzahl an möglichen Verhaltensabläufen, so z. B. bei nebenläufigen Systemen, in denen mehrere Systemkomponenten unabhängig voneinander gleichzeitig Systemaufgaben bearbeiten. Eine weitere Form der Komplexität ergibt sich aus der Notwendigkeit, ein großes System in Teilsysteme zu zerlegen, wodurch sich zwangsläufig Schnittstellen zwischen den Teilsystemen ergeben, die entwickelt und unterhalten werden müssen. Sowohl Nebenläufigkeit wie auch Systemdekomposition führen zu einer kombinatorischen, exponentiellen Komplexitätsexplosion.

Die Beherrschung dieser Komplexität ist eine Zentralaufgabe des Software-Engineerings, wenn nicht gar der Informatik. Die objektorientierte Modellierung unterstützt daher eine modulare Dekomposition des Systems in Teilsysteme, den Umgang mit nebenläufigen Systemausführungen wie auch die Modellierung von Schnittstellen zwischen Teilsystemen. Ein weiterer Mechanismus mit dem Ziel des Umgangs mit Komplexität ist das oben bereits beschriebene Konzept der Verallgemeinerung, Verfeinerung und Vererbung von Objektstrukturen. Wenngleich damit

die intrinsische Komplexität von Systemen nicht vermieden werden kann, so kann damit doch der Umgang mit Komplexität vereinfacht oder sogar erst ermöglicht werden.

3 „Unvollständigkeit“ des expliziten Rechts (implizites Wissen und Notwendigkeit der Konkretisierung)

Selbst wenn die Komplexität grundsätzlich beherrscht werden kann, besteht bei der Modellierung des Rechts die Herausforderung, dass es vielfach auch auf implizitem Wissen beruht, dessen Explikation eine nicht zu unterschätzende konzeptionelle Aufgabe für die Rechtswissenschaft ist. Im Bereich der expliziten Programmierung müssten nicht nur alle entschiedenen, sondern auch im Vorhinein alle zu entscheidenden Fallkonstellationen berücksichtigt und die einzelnen Schritte der Rechtsanwendung bis in das letzte Detail entsprechend festgelegt werden, was nur für gleichartige Rechtsverhältnisse mit geringer Varianz praktikabel erscheint.²⁵ Der Einsatz maschinellen Lernens setzt wiederum voraus, dass hinreichende Trainingsdaten existieren, die jedoch in aller Regel nur die bisher entschiedenen Konstellationen umfassen.²⁶ Vor allem aber beruhen die rechtlichen Regelungen auf Rechtsbegriffen und Tatbeständen, die unscharf und in Bedeutung und Reichweite entwicklungsfähig sind, so dass ihnen eine Vagheit inhärent ist und sie der Rechtsanwendung eine konkretisierende Funktion zuweisen, die aus Sicht algorithmenbasierter Anwendungen eine strukturelle Schwäche darstellt, während sie für die Entwicklungsoffenheit des Rechts zentral ist.²⁷ Dies gilt insbesondere, aber nicht nur für die unbestimmten Rechtsbegriffe. Hier spiegeln sich die oben angesprochenen Stärken und Schwächen der Begriffsjurisprudenz und der Wertungsjurisprudenz.

4 Konsequenz: approximative Modellierung

Die Rechtsmodellierung beruht bisher häufig auf Entscheidungsbäumen, die Voraussetzungen über Und-/Oder-Verbindungen abbilden.²⁸ Sie strebt dabei weitgehende Vollständigkeit an.

²⁵ Vgl. *Broemel*, in: Kuhli/Rostalski (Fn. 14), S. 123, 143 ff.

²⁶ Vgl. *Broemel*, in: Kuhli/Rostalski (Fn. 14), S. 146 f.

²⁷ Vgl. *Broemel*, in: Kuhli/Rostalski (Fn. 14), S. 147 ff. m. Verw. auf *Luhmann*, *Recht und Automation in der öffentlichen Verwaltung*, 2. Aufl. 1997, S. 52 ff., 58 f.

²⁸ Vgl. etwa *Breidenbach*, in: Breidenbach/Glatz (Fn. 9), S. 340 ff.

Aus der Sicht des Rechts ist eine vollständige Modellierung bis ins letzte Detail jedoch aus den genannten Gründen – selbst wenn sie leistbar wäre – sehr problematisch. Vorzugswürdig – und wohl auf absehbare Zeit auch realistischer – scheint eine Modellierung, die nur bis zu einem gewissen Grad erfolgt und den Rechtsanwender unterstützt, insbesondere bei der Identifikation von problematischen Stellen, ihm aber die Rechtsanwendung im Sinne der Konkretisierung auch konzeptionell überlässt.

Die approximative Modellierung und Analyse ist ein auch in der Softwareverifikation wohl bekanntes Konzept. Das ursprüngliche Ziel der formalen Modellierung und Analyse von Software war es, die Korrektheit der Software unter allen Ausführungsvarianten mathematisch zu beweisen. Dieses Ziel wird auch als vollständige Softwareverifikation bezeichnet. In der praktischen Anwendung zeigt sich jedoch, dass reale, nicht-triviale Softwaresysteme eine Komplexität besitzen, die es besonders für automatisierte Verifikationstechniken, wie z.B. dem „Model Checking“ (Modellprüfverfahren), unmöglich macht, eine vollständige Softwareverifikation durchzuführen. Allerdings zeigt es sich, dass die Verwendung von automatischen Verifikationsverfahren auch bei Aufgabe der Vollständigkeit praktisch sehr wertvolle Informationen liefern kann. Auch wenn nicht alle Fehler in einem Softwaresystem gefunden werden können, so kann ein unvollständiges Verfahren doch Fehler finden und damit Anhaltspunkte für die Fehlerbehebung in der Software („Debugging“) liefern. Einen solchen Ansatz bezeichnet man auch als Falsifikation. Als besonders effektiv haben sich dabei Verfahren erwiesen, die das Verhalten des Systems durch Analyse einer Abstraktion unter- oder überapproximieren und dadurch auf komplexe Systeme mit großem Zustandsraum anwendbar sind. Wenn in einem solchen Verfahren eine Falsifikation in dem abstrakten Modell erreicht wird, dann kann dies einen Fehler in dem konkreten Modell repräsentieren und es kann somit eine Warnung („Red Flag“) an den Benutzer ausgegeben werden, dass ein möglicher Fehler vorliegt.

Auch in unserer Modellierung und Analyse von Unternehmenskaufverträgen verwenden wir eine Unterapproximation mit dem Ziel der Falsifikation. Unser Ziel ist es also nicht, die Korrektheit bzw. Fehlerfreiheit eines Vertragswerks zu beweisen, sondern ein Warnsystem bereitzustellen, das im Sinne des „Red Flag“-Ansatzes auf das mögliche Erreichen inkonsistenter Zustände des Vertrags oder inkonsistente Vertragsausführungen hinweisen kann. Es ist dabei möglich, dass wir nicht alle möglichen Ausführungen eines Vertrags erfassen. So können z.B. die Folgen von Gerichtsentscheidungen bezüglich einzelner Vertragsklauseln oder der Vertragsgrundlagen möglicherweise in unserem Modell nicht abgebildet sein. Dennoch kann das System Ausführungen des Vertrags illustrieren, die z.B. die Unmöglichkeit der Durchsetzung eines Anspruchs oder die Ausführung eines Vertrags ohne Ausführung eines einzigen

Primäranspruchs verdeutlichen. Diese sollten den Benutzer dazu ermuntern, die entsprechenden Vertragsbestimmungen kritisch zu hinterfragen.

Bei unserer Analyse bedienen wir uns der oben angesprochenen SMT-Technologie, für die es seit etwa 15 Jahren sehr effiziente, allgemein verfügbare, das automatische Schließen unterstützende Softwarewerkzeuge gibt.

VII Logik als Fundament der Modellierung

1 Abgrenzung von Methoden maschinellen Lernens

Die Art des logischen Schließens, das wir hier vorschlagen, beruht auf formaler mathematischer Logik und den für die von uns verwendeten Logiken entwickelten Entscheidungsprozeduren. So können wir z. B. Ausdrücke der ganzzahligen linearen Arithmetik verwenden, um die Einhaltung von zeitlichen Beschränkungen für die Erfüllung von Ansprüchen auszudrücken und deren Konsistenz zu überprüfen. Die Schlussmechanismen arbeiten gänzlich auf Basis einer symbolischen Repräsentation der Vertragsinhalte und der Konsistenzbedingungen.

Im Bereich der Künstlichen Intelligenz haben sich in der jüngeren Vergangenheit wissensverarbeitende Systeme auf Grundlage der sub-symbolischen Künstlichen Intelligenz etabliert. Beispiele sind besonders die in der jüngeren Vergangenheit entwickelten Spracherzeugungssysteme, die auf durch maschinelles Lernen erzeugten großen Sprachmodellen basieren, wie z. B. ChatGPT-4 oder auch LawGPT. Diese Systeme erzeugen natürlichsprachliche Ausgaben für gegebene Eingabeabfragen basierend auf rein statistischen Sprachmodellen. Entsprechend sind die Ausgaben lediglich statistisch zu validieren und bieten keine präzisen und in jedem Fall verlässlichen ja/nein Antworten. In der KI-Forschung wird derzeit stark diskutiert, ob diese Systeme intelligent sind, und ob sie zu logischem Schließen befähigt sind.²⁹ Wir beschränken uns in dieser Arbeit auf die Verwendung der symbolischen Techniken, aber die Betrachtung der sub-symbolischen Techniken in der Analyse juristischer Artefakte wird Gegenstand zukünftiger Forschung sein.

²⁹ Bubeck u. a., Sparks of Artificial General Intelligence: Early experiments with GPT-4, 13.4.2023, <https://doi.org/10.48550/arXiv.2303.12712> (4.2.2024).

2 Klassische Prädikatenlogik erster Stufe vs. Deontische Logik

In der logischen Modellierung von juristischen Artefakten kommt der Deontischen Logik und dem nichtmonotonen Schließen eine besondere Bedeutung zu. Die Deontische Logik erlaubt die Formulierung von normativen Aussagen. Dies sind Aussagen, die sich auf das Sollen beziehen und somit Konzepte wie Pflichten, Gebote, Verbote und Erlaubnisse zu formulieren erlauben. In der Deontischen Logik werden somit Operatoren für Gebot (A), Verbot (E), Erlaubnis (O) und nicht geboten sein (I) eingeführt, die auf Propositionen und aussagenlogische Formeln angewandt werden können, woraus wahre oder falsche Aussagen entstehen können. Die Semantik dieser Operatoren wird durch deren Beziehungen erklärt, wobei diese Widersprüchlichkeiten innerhalb einer Formel zur Folge haben können. Die Deontische Logik ist somit weder widerspruchsfrei noch monoton, woraus sich Probleme für die Entscheidbarkeit der Erfüllbarkeit einer Formel ergeben.³⁰

In dem von uns entwickelten Modellierungs- und Analyseansatz verwenden wir zur Modellierung der logischen Abhängigkeiten zwischen Ausführungsschritten eines Vertrags nicht-deontische entscheidbare Quantoren-freie Fragmente der Prädikatenlogik erster Stufe, genannt Theorien. In unserem Ansatz verwenden wir speziell die Theorie der linearen Arithmetik, um zeitliche Abhängigkeiten sowie die Berechnung von Geldbeträgen (Kaufpreis, Entschädigung im Falle einer Gewährleistung, Kompensation bei Rückabwicklung des Vertrags) zu modellieren. Ferner werden die Theorien der Uninterpretierten Funktionen und der Funktionellen Konsistenz verwendet, um z.B. Eigentumsverhältnisse zu modellieren. Die Entscheidbarkeit dieser Theorien bedeutet, dass es automatische und vollständige Algorithmen gibt, die die Erfüllbarkeit einer Konjunktion von logischen Fakten in einem Vertrag berechnen.

Zur Entscheidung der Erfüllbarkeit werden für eine betrachtete Theorie in der Praxis effiziente Algorithmen verwendet. So findet z.B. der entscheidbare Test auf Existenz eines erfüllenden Modells aus dem allgemeinen Simplexverfahren Verwendung bei der Bestimmung der Erfüllbarkeit einer Formel in der Theorie der linearen reellen Arithmetik. Für die unterschiedlichen entscheidbaren Theorien werden Implementierungen der Entscheidungsverfahren in den oben bereits genannten SMT-Solvern bereitgestellt. In der Praxis besonders hilfreich sind dabei zwei von SMT-Solvern bereitgestellte Funktionen. Falls eine Formel erfüllbar ist, wird ein konkretes, erfüllendes Modell berechnet und ausgegeben. Dies bedeutet z.B., dass dem Benutzer eine konsistente Ausführung eines Vertrags präsentiert

³⁰ *Prisacariu/Schneider*, A dynamic deontic logic for complex contracts, *Journal of Logical and Algebraic Methods in Programming* 2012, 458 ff.

werden kann. Sollte eine Formel von dem SMT-Solver als nicht erfüllbar berechnet werden, dann berechnet er einen Nichterfüllbarkeits-Kern, wobei es sich um eine minimale Anzahl an Teilformeln handelt, die zueinander im Widerspruch stehen. Ähnlich dem Vorgehen in der Softwareentwicklung kann dieser Nichterfüllbarkeits-Kern dazu verwendet werden, das analysierte Vertragswerk zu „debuggen“, also die Ursachen der Inkonsistenz zu ermitteln und diese zu beheben.

Gegenüber den Fragmenten der Logik erster Stufe, die wir in unserer Arbeit verwenden, hat die Deontische Logik den Vorteil, dass sie die Formulierung normativer Aussagen ermöglicht. Allerdings sind diese für die von uns betrachtete Anwendung nicht von Bedeutung, da wir lediglich die Konsistenz von durch zweitwertige Logik beschreibbaren Vertragsklauseln überprüfen wollen. Dadurch können wir auf die Möglichkeiten des automatischen Schließens auf Grundlage der von uns verwendeten Formeln der Prädikatenlogik erster Stufe zurückgreifen.

Ein weiterer Vorteil der von uns verwendeten Logik ist die Möglichkeit, Aussagen über unterschiedliche Domänen, die sogenannten Theorien, zu formulieren. So können wir in der gleichen logischen Formalisierung Daten, Kaufpreise, Eigentumsverhältnisse oder, bei Verwendung eines geeigneten SMT-Solvers, auch Zinsberechnungen in die Modellierung einbeziehen. Eine entsprechende Erweiterung von Fragmenten der Deontischen Logik zu „Deontic Satisfiability Modulo Theories“ ist uns nicht bekannt.

VIII Fallstudie: Modellierung und Konsistenzanalyse eines Unternehmenskaufvertrags

1 ContractCheck

Im Folgenden demonstrieren wir die Modellierung eines Vertrags am Beispiel unseres aktuellen Projekts ContractCheck³¹. Die Idee hinter dem Projekt ist es, einen Unternehmenskaufvertrag mit Hilfe unseres Werkzeugs ContractCheck, das intern SMT-Solver aufruft, auf Inkonsistenzen zu überprüfen. Dies erfordert die Modellierung der vertraglichen Regelungen in entscheidbaren Fragmenten der Prädika-

³¹ Khoja/Kölbl/Leue/Wilhelmi, Automated Consistency Analysis for Legal Contracts, in: Model Checking Software. SPIN 2022. Lecture Notes in Computer Science, 2022, S. 1 ff.

tenlogik erster Stufe. Unsere Modellierung ist objektorientiert und erfolgt mit Hilfe von Klassen- und Objektdiagrammen in der UML.³²

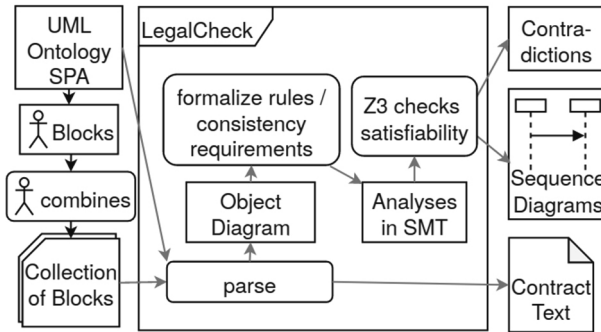


Abb. 1: Struktur des ContractCheck-Ansatzes und -Werkzeugs.

Abb. 1 präsentiert die Struktur des ContractCheck-Werkzeugs. Basierend auf einer UML-Ontologie für Unternehmenskaufverträge werden strukturierte, parametrisierte Textblöcke entwickelt, die der Benutzer anhand des Vertragstexts kombinieren kann. Innerhalb der Software wird diese Menge der übergebenen Textblöcke intern in eine logische Repräsentation gebracht und das entstehende logische Nebenbedingungssystem mithilfe des SMT Solvers Z3³³ auf Inkonsistenzen untersucht. Im positiven Fall werden erfüllbare Vertragsausführungen als Sequenzdiagramme ausgegeben, im Falle von detektieren Inkonsistenzen werden diese als „Red Flag“-Annotationen an den involvierten Textblöcken ausgegeben. Sämtliche der in Abb. 1 innerhalb der Blocks „LegalCheck“ angeführten Verarbeitungsschritte erfolgen vollautomatisch.

2 Modellierung des Vertrags

Ausgangspunkt ist die Erfassung der möglichen Regelungen einschließlich ihrer Beziehungen untereinander in entsprechenden Verträgen in abstrakter Form als Klassen in einem Klassendiagramm.

³² Vgl. zum Vorgehen *Soltana* u. a., in: Dingel u. a. (Hrsg.), *Model-driven engineering languages and systems*, 2014, S. 450 ff.

³³ *de Moura/Bjørner*, *Z3: An Efficient SMT Solver*, in: *TACAS 2008. Lecture Notes in Computer Science*, 2008, S. 337 ff.

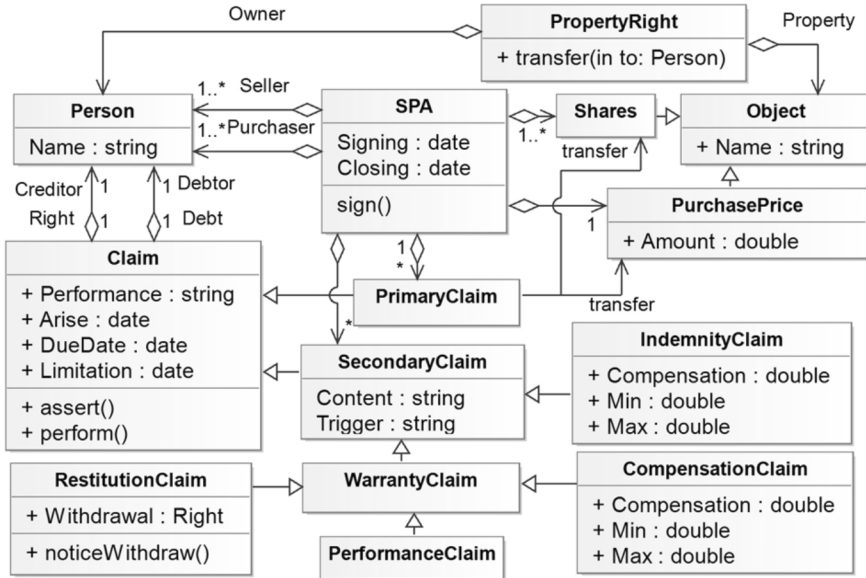


Abb. 2: Klassendiagramm eines Unternehmenskaufvertrags.

Dabei hat jede Klasse einen Namen (oberes Feld), kann ein oder mehrere Attribute (mittleres Feld) und eine oder mehrere Operationen (unteres Feld) haben. Die Beziehungen zwischen den Klassen sind als Generalisierung (Pfeil mit Dreieck) bzw. Spezialisierung (umgekehrter Pfeil mit Dreieck) und als Aggregation als eine „besteht aus“-Beziehung (Raute mit Pfeil) dargestellt. Die einfachen Pfeile stellen sonstige Beziehungen (Assoziationen) dar. Im Beispiel besteht der Unternehmenskaufvertrag (SPA) aus Primäransprüchen (**PrimaryClaim**), die eine Spezialisierung der allgemeinen Ansprüche (**Claim**) darstellen und auf die Übertragung der Anteile (**Shares**) und des Kaufpreises (**PurchasePrice**) gerichtet sind. Bei der Spezialisierung werden die Attribute und Operationen von der generellen Klasse auf die spezielle Klasse vererbt, wenn sie dort nicht überschrieben werden. Die spezielle Klasse hat also grundsätzlich die gleichen Attribute und Operationen wie die generelle Klasse, kann diese aber anders definieren. Dies entspricht der Funktion von Allgemeinen und Besonderen Teilen im Recht, wo die Regelungen im Allgemeinen Teil auch für die Regelungen im Besonderen Teil gelten, soweit diese nicht speziellere Regelungen enthalten, die die Regelungen des Allgemeinen Teils verdrängen.

Dieses Klassendiagramm kann dann mit den Regelungen eines konkreten Kaufvertrags gefüllt werden und wird so zu einem Objektdiagramm, wie in Abb. 3 illustriert.

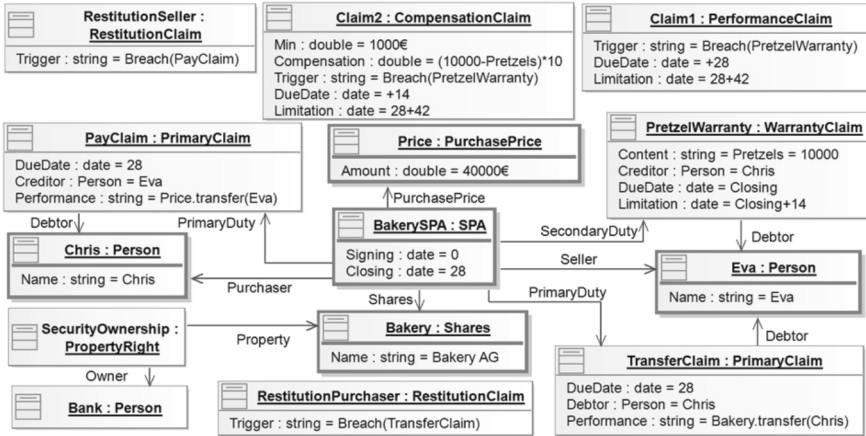


Abb. 3: Objektdiagramm über den Kaufvertrag zwischen Eva und Chris über eine Bäckerei.

Die Beziehungen zwischen den einzelnen Objekten ergeben sich dabei aus dem Klassendiagramm. Das Objektdiagramm zeigt die Belegung der Attribute der einzelnen Klassen im konkreten Kaufvertrag.

Die Erfassung der Regelungen des konkreten Vertrags erfolgt bei ContractCheck in der bisherigen Version durch den Benutzer, der die Regelungen eines Vertrags in Text- bzw. Regelungsblöcken, bei denen es sich um parametrisierte strukturierte Sprachblöcke handelt, übersetzt.

3 Analyse der Regelungen des Vertrages

Das Objektdiagramm lässt sich in maschinenlesbarer Logik ausdrücken, die dann durch SMT-Solver auf Inkonsistenzen überprüft wird. So kann etwa erkannt werden, wenn die für einen Schadensersatzanspruch geltende Verjährungsfrist ab Vertragserfüllung (Closing) kürzer ist als Benachrichtigungs- plus Abhilfefrist. Ähnlich könnte erkannt werden, wenn Abweichungen hinsichtlich der Produktionskapazität sowohl zu einer Anpassung des Kaufpreises als auch zu einer Zahlung aufgrund einer Garantie führen und diese zu einer Doppelung oder einer Neutralisierung.

Zur Konkretisierung der Möglichkeiten der logischen Konsistenzanalyse von Regelungen des Vertrags betrachten wir die Modellierung und Analyse des Übertragungsanspruchs für die Brezelbäckerei. Die Übertragung der Bäckerei an die Bank als Sicherheit für eine Hypothek wird durch die Formel $owner(Bakery) = Bank$ modelliert. Eine ganzzahlige Variable d_u bezeichnet den Tag

nach Vertragsunterzeichnung, an dem die Übertragung stattfindet. Falls die Übertragung nicht stattfindet, hat d_u den Wert -1 . Die Vertragsbedingung, dass die Übertragung 28 Tage nach Vertragsunterzeichnung stattfinden muss und dass es eine Voraussetzung ist, dass Eva tatsächlich Eigentümerin der Bäckerei ist, wird durch die Formel $28 \leq d_u \wedge \text{owner}(\text{Bakery}) = \text{Eva}$ ausgedrückt. Insgesamt werden die Konsistenzbedingungen an die Übertragung des Vertragsgegenstands durch die Formel $\text{owner}(\text{Bakery}) = \text{Bank} \wedge (d_u = -1 \vee (28 \leq d_u \wedge \text{owner}(\text{Bakery}) = \text{Eva}))$ ausgedrückt. Es ist zu beachten, dass es sich bei *owner* um eine nicht interpretierte Funktion handelt. Der durch ContractCheck aufgerufene SMT-Solver detektiert hier eine funktionelle Inkonsistenz (eine Funktion muss bei Aufruf mit dem gleichen Argument immer das gleiche Resultat liefern) und liefert einen Unerfüllbarkeitskern, der durch ContractCheck unter Verweis auf die betreffenden Blöcke dargestellt wird, siehe Abb. 4.

Eine weitere Analyse betrachtet die Möglichkeit der Ausführung der in einem Vertrag enthaltenen Ansprüche, speziell den Übertragungsanspruch, Schadenersatzansprüche für Verkäufer und Käufer, den Zahlungsanspruch, einen Gewährleistungsanspruch sowie Kompensationsansprüche. Die Fähigkeit des SMT-Solvers einzelne Teilformeln mit Priorität zu erfüllen, erlaubt es zu analysieren, ob eine Vertragsausführung unter Erfüllung von Primäransprüchen möglich ist. In dem Fall der Bäckerei ist dies nicht der Fall, da der Übertragungsanspruch aufgrund der oben dargestellten Inkonsistenz nicht ausführbar ist. Der SMT-Solver berechnet dennoch eine Ausführung des Vertrags, bei der die Übertragung der Bäckerei nicht ausgeführt wird, der Kaufpreis dennoch gezahlt wird und der Käufer dann für den Kaufpreis durch Rückzahlung entschädigt wird. Das entsprechende Ausführungsszenario wird durch ein Sequenzdiagramm von ContractCheck visualisiert.

Für eine detaillierte Beschreibung der Modellierung und Analyse verweisen wir den Leser auf unser technisches Papier³⁴.

4 „Red Flag“-System

Die Modellierung und das darauf aufbauende Programm leisten also keine vollständige Überprüfung des Vertrages, sondern können nur die in der Modellierung erfassten Informationen auf Inkonsistenzen prüfen. Ob tatsächlich eine Inkonsistenz vorliegt, insbesondere ob diese nicht mit Hilfe des Gesetzesrechts oder der ergänzenden Vertragsauslegung vermieden wird, kann nur der Rechtsanwender feststellen. ContractCheck unterstützt ihn jedoch bei der Suche nach Inkonsistenzen

³⁴ Khoja/Kölbl/Leue/Wilhelmi, in: SPIN 2022 (Fn. 31), S. 1 ff.

in unübersichtlichen und mit verschiedenen Personen verhandelten und mehrfach geänderten Verträgen, indem es ihm „Red Flags“ anzeigt, wo es Inkonsistenzen sieht.



Abb. 4: „Red Flag“ für Inkonsistenz zweier Vertragsblöcke.

Abb. 4 illustriert die Ausgabe von ContractCheck, mit der die Software auf die oben diskutierte Inkonsistenz zwischen zwei Bestimmungen des Vertrags, welche die Eigentumsrechte an der Bäckerei betreffen, hinweist. In Abb. 5 wird die Ausführung des Kaufvertrags in der Form eines UML-Sequenzdiagramms in dem oben geschilderten Fall dargestellt, in der der Kaufgegenstand nicht übertragen wird und es zu einer Entschädigung des Verkäufers für den gezahlten Kaufpreis kommt. ContractCheck illustriert also nicht nur Inkonsistenzen, sondern auch mögliche Vertragsausführungen. Das Diagramm illustriert eine mögliche Ausführung des Vertrags, selbst wenn dieser die oben genannte Inkonsistenz enthält und daher der Primäranspruch der Übertragung des Verkaufsgegenstands nicht erfüllbar ist.

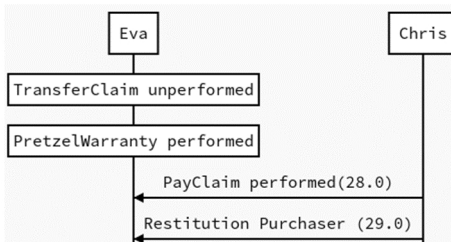


Abb. 5: Sequenzdiagramm für eine erfolgreiche Vertragsausführung.

IX Fazit

Als Fazit lässt sich festhalten, dass auch im Recht modelliert wird und Methoden der Modellierung aus der Informatik auf das Recht bzw. die Rechtswissenschaft übertragbar sind. Die objektorientierte Modellierung der Informatik weist etwa Parallelen zu der Systembildung durch dogmatische Figuren im Recht auf. Oder, wie eine Doktorandin der Informatik als Fazit eines Kurses zum Vertragsrecht für Nichtju-

risten formulierte: Recht ist wie Informatik, nur mit viel mehr und viel schlechter definierten Variablen.

Demgemäß lässt sich Recht auch mit Hilfe der Methoden der Informatik modellieren – allerdings nur zu einem gewissen Grad. Eine vollständige Modellierung des Rechts erscheint unrealistisch. Ihr steht nicht nur die Komplexität des Rechts entgegen, sondern auch seine inhärente „Unvollständigkeit“, die Voraussetzung der Entwicklungs Offenheit des Rechts ist, aber die Notwendigkeit der Konkretisierung zur Folge hat.

Modellierung ist auch Voraussetzung für die (teilweise) Automatisierung der Rechtsanwendung. Die Wahl der Methode der Modellierung hängt – wie immer – vom jeweiligen Zweck ab. Deontische Logik kann etwa normative Konzepte wie Pflichten, Gebote, Verbote und Erlaubnisse besser ausdrücken. Klassische Prädikatenlogik erster Stufe ist demgegenüber besser geeignet, entscheidbare Formeln als Grundlage des automatischen Schließens zu modellieren. Sie erlaubt auch eine ontologische Modellierung von Verträgen und der Zustände ihrer Ausführung mit Hilfe von UML.

Modellierung (und damit auch Automatisierung) des Rechts ist demgemäß nur bis zu einem gewissen Grad möglich und praktikabel, dort aber nützlich. Sie kann menschliche Rechtsanwendung ergänzen und unterstützen, aber nur in eng begrenzten Ausnahmefällen ersetzen. Statt auf eine vollständige Modellierung und Automatisierung des Rechts zu setzen, empfiehlt sich daher in der Regel ein pragmatischer Ansatz.

In unserer Fallstudie haben wir gezeigt, dass in Analogie zu den Methoden der automatischen Softwareverifikation die Techniken der algorithmisch entscheidbaren, über unterschiedliche Theorien definierten Logiken erfolgreich auch in der Konsistenzanalyse für juristische Verträge verwendet werden können. Von zentraler Bedeutung für die praktische Anwendbarkeit ist dabei die Verwendung automatisierter logischer Methoden, welche die Entwicklung vollständig automatisierter Werkzeuge ermöglichen. Die Idee ist dabei, das Formale an der Methode im Hintergrund zu halten und dem Benutzer lediglich „Red Flags“ oder andere qualitative Diagnoseinformationen zur Verfügung zu stellen. Denkbar wäre auch ein Tool entsprechend der Prüfung von Rechtschreibung und Grammatik in Word.

Während wir bisher durch einen eigenen Vertragsgenerator erzeugte Verträge analysiert haben, arbeiten wir momentan an einer Erfassung bestehender Verträge mittels Texterkennung, die die Objekte als befüllte Klassen erkennen soll. Dies würde die Analyse auch bestehender Verträge ermöglichen.

Unser Fazit: Not all of law is logic, but some of it is!

