

Wann ist ein juristischer Text strukturiert?

—

Die Sicht der Informatik und der Künstlichen Intelligenz

Michael Kohlhasse
Informatik, FAU Erlangen-Nürnberg
michael.kohlhasse@fau.de

9. Januar 2022

Zusammenfassung

In der Diskussion zur Digitalisierung von Zivilprozessen wird oft das Konzept von „strukturierten Dokumenten“ als Lösung gehandelt. Hier gehen wir der Frage nach, wann *aus der Sicht der Informatik und Künstlichen Intelligenz* ein Dokument als „strukturiert“ angesehen werden sollte. Wir interpretieren „strukturierte Dokumente“ als flexibel tief/breit formalisierte Dokumente und untersuchen den Zusammenhang zwischen Formalisierungsgrad und Ertrag der auf die Formalisierung/Strukturierung basierten Mehrwertdienste.

1 Einleitung

Der Großteil der juristischen Arbeits- und Kommunikationsprozesse im Zivilprozess wird über spezielle Dokumente vollzogen; insbesondere Klageschrift, Klageerwiderung, Replik und Duplik. Daher wird die Digitalisierung des Zivilprozesses wohl vor Allem über die Digitalisierung von Dokumenten aller Arten geschehen.

Die Tagung „*Digitalisierung von Zivilprozess und Rechtsdurchsetzung*“ die in diesem Band dokumentiert wird wurde durch das Diskussionspapier „*Modernisierung des Zivilprozesses*“ [Ziv21] der Arbeitsgruppe „Modernisierung des Zivilprozesses“ im Auftrag der Präsidentinnen und Präsidenten der Oberlandesgerichte, des Kammergerichts, des Bayerischen Obersten Landesgerichts und des Bundesgerichtshofs motiviert. Dieses Papier untersucht wie durch eine Digitalisierung der Dokumente und Kommunikationsprozesse Zivilprozesse bürgerfreundlicher, effizienter und ressourcenschonender gestaltet werden können. So soll perspektivisch beispielsweise als Grundlage jedes Zivilprozesses ein gemeinsames Basisdokument entstehen das Kläger- und Beklagtenvortrag zum Lebenssachverhalt festhält

und durch seine inhaltliche Struktur (Relationentabelle) die Verhandlung abbildet und vereinfacht.

Eine Studie des „Nationale E-Government Kompetenzzentrums“ (NEGZ) geht sogar noch weiter: [Rum+] schlägt vor domänenspezifische Sprachen (DSL) und formale Domänenmodelle zu entwickeln, um Gesetzestexte maschinell umsetzbar zu machen und sie computerunterstützt handhaben, aktualisieren und auf konkrete Sachverhalte anwenden zu können.

In vielen Beiträgen dieses Bandes wird aus *Sicht der juristischen Praxis* diskutiert, wie die im Rahmen von Zivilprozessen verwandten Dokumente digitalisiert und strukturiert werden können, so dass sich Digitalisierungsgewinne in deren Nutzung und Verarbeitung gewinnen lassen. Komplementär dazu geht dieses Übersichts- und Positionspapier der Frage nach, wann *aus der Sicht der Informatik und Künstlichen Intelligenz*, also der anderen wichtigen Disziplin in der Digitalisierung von Zivilprozessen ein Dokument als „strukturiert“ angesehen werden sollte und so einen Beitrag zur Diskussion in diesem Band liefern. Wir verwenden hier den Begriff „strukturiert“ statt dem in der Informatik gebräuchlicheren Begriffs „formal“, da in [Ziv21] von „strukturierten Prozessen“ und „strukturierten Dokumenten“ als Grundlage von Automatisierung und Digitalisierung von Zivilprozessen die Rede ist.

Im Prinzip ist die Grundantwort der Informatik als Strukturwissenschaft der Automatisierung von Prozessen sehr einfach:

***Antwort:** Ein Dokument ist im Sinne der Informatik als „formal“ oder „strukturiert“ anzusehen, wenn die darin enthaltene Information so explizit gemacht wurde, dass sie maschinenverarbeitbar ist, so dass die Verarbeitungsprozesse, die auf diesem Dokument aufbauen, maschinell unterstützt oder sogar automatisiert werden können.*

Der Begriff „formal“ basiert auf der Erkenntnis der klassischen/griechischen Philosophie, dass manchmal die Gültigkeit einer Aussage oder eines Arguments alleine aus seiner Form abgeleitet werden kann. Wir nennen eine solche Aussage oder Argument dann **formal**; entsprechend bezeichnen wir den Prozess, eine Aussage oder Argument in eine formale Form zu bringen - normalerweise durch Explizierung impliziter Teile - als **Formalisierung**. Mit dem Aufkommen elektronischer Rechenmaschinen im 20. Jahrhundert wurde diese Erkenntnis erweitert dahingehend, dass formale Dokumente (also z.B. Programme und Datensätze) zur Automatisierung der Datenverarbeitung eingesetzt werden können. Dies ist die Grundlage der modernen Informatik; die Künstliche Intelligenz erweitert diesen Ansatz auf das „Rechnen mit Dokumenten-Inhalten: z.B. Wissen und Argumente“. Die Antwort oben ist einfach eine Ausformulierung der Kombination dieser Ideen.

Es soll in diesem Artikel primär um die natürliche Folgefrage gehen: *Was bedeutet das für die juristische Dokumente und Verarbeitungsprozesse?* Dabei soll vor allem betrachtet werden, was die Voraussetzungen für eine maschinelle Verarbeitbarkeit von Dokumenten sind, und wie dieses mit der Natur der juristischen Verarbeitungsprozessen wechselwirkt. Insbesondere bemerken wir, dass in der Grundantwort oben der Grad der Explizierung

sonst implizit in den Dokumenten enthaltenen Wissens von den intendierten Verarbeitungsprozessen abhängen kann. Auf diesen Zusammenhang wollen wir in diesem Artikel ein besonderes Augenmerk richten.

In Abschnitt 2 legen wir die Basis für die Diskussion indem wir zwischen natürlichen und formalen Sprachen unterscheiden. In Abschnitt 3 weichen wir diese Unterscheidung auf um zu einem realistischeren Begriff für „strukturierte Dokumente“ zu kommen um dann in Abschnitt 4 den Zusammenhang zwischen Formalisierungs/Strukturierungsgrad und dem Ertrag von einigen einschlägigen Mehrwertdiensten zu untersuchen. Abschnitt 5 beschließt den Artikel.

2 Natürliche und Formale Sprachen

Eine grundlegende Voraussetzung für die maschinelle Informationsverarbeitung ist die Verwendung von *formalen Sprachen*. Während juristischen Dokumente in einer Fachsprache - einer Spielart der *natürlichen Sprache*, die durch Fachbegriffe, spezielle Idiome und Sonderbedeutungen angereichert ist - verfasst sind, müssen Dokumente für die maschinelle Verarbeitung zu „Daten“ werden. Definitionsgemäß besteht eine **formale Sprache** aus einer Menge „erlaubter“ Zeichenketten (die **Syntax** der Sprache; z.B. Sätze oder Texte), denen dann systematisch eine Bedeutung zugeordnet werden kann (die **Semantik**, z.B. in Form von Maschineninstruktionen). Die Informatik verwendet viele Tausende formaler Sprachen für verschiedenste Anwendungen, z.B. Programmiersprachen zur Beschreibung von Berechnungsprozessen, Dokumentenbeschreibungssprachen für die Struktur von Texten, Audio- und Video-Aufnahmen, so wie Logiken für die Beschreibung und Automatisierung von Wissens- und inferenzbasierten Denkprozessen.

Am Besten ist der Gegensatz zwischen formalen und natürlichen Sprachen vielleicht an einem einfachen Beispiel zu verstehen: Die (natürlichsprachliche) Anfrage (1) muss zur maschinellen Bearbeitung z.B. in eine Datenbankabfrage der Form (2) gebracht werden:

1. Wann geht der nächste Zug nach Frankfurt?
2. `SELECT * FROM Verbindung WHERE Ziel='Frankfurt'`

Bei (2) handelt es sich um einen Ausdruck der formalen Sprache SQL, die von einem relationalen Datenbanksystem verarbeitet werden kann. Dieses analysiert die syntaktische Struktur der Anfrage und übersetzt diese in eine Abfolge von Maschinenbefehlen (semantische Analyse) die auf dem Microchip eines Computers ausgeführt werden und aus einer Datenbanktabelle **Verbindung** alle Verbindungen mit Ziel **Frankfurt** herausuchen. Ohne eine maschinenverarbeitbare Syntax und Semantik der (formalen) Sprache SQL wäre diese Verarbeitung unmöglich. Wir sehen also im Sinne der Definition oben, das „Dokument“ (2) als strukturiert an, (1) dagegen aber nicht, da die natürlichsprachliche Anfrage (bisher¹) nicht maschinell verarbeitet werden kann.

¹Dem Einwand „Aber meine Alexa versteht (1) doch ohne weiteres, also muss das doch auch formal sein.“ sei hier dadurch begegnet, dass ein einzelner maschinenverarbeitbarer Satz noch keine formale Sprache aus-

Dieses einfache Beispiel zeigt schon zwei wichtige Unterschiede zwischen natürlichen und formalen Sprachen, die wir in Abb. 1 zusammenfassen: Natürliche Sprachen haben sich über lange Zeiträume für die Kommunikation zwischen Menschen entwickelt. Sie nutzen die kognitiven Fähigkeiten des Hörers/Lesers um komplexe Bedeutungen effizient und flexibel zu vermitteln.

	Formale Sprache	Natürliche Sprache
Paradigma	maschinenorientiert	menschenorientiert
Syntax	rigide	flexibel
Bedeutung	wohldefiniert, eindeutig	gebrauchsdefiniert, mehrdeutig

Abbildung 1: Formale vs. Natürliche Sprachen

Im Gegensatz dazu sind formale Sprachen wie SQL von Menschen entworfen worden, um mit Computern zu kommunizieren. Der Computer hat (trotz der Bemühungen der Künstlichen Intelligenz in den letzten 70 Jahren) keine höheren kognitiven Fähigkeiten. Daher werden Syntax und Bedeutung (Semantik) formaler Sprachen in (mathematisch exakten) Spezifikationen definiert und dann in informationsverarbeitenden Systemen implementiert. Um die Implementation möglich zu machen müssen sowohl die syntaktische Wohlgeformtheit als auch die intendierte Bedeutung eines Ausdrucks so streng gefasst sein, dass (relativ einfache) Programme sie eineindeutig erfassen können.

Für flexible Syntax - „... der Gersch, der wo in der Hauptstraß wohnt ...“ - oder Mehrdeutigkeiten - bezieht sich das Wort „Bank“ in „Peter setzte sich auf die Bank“ auf ein Sitzmöbel oder ein Finanzinstitut? - ist in Computersprachen kein Platz. In natürlichen Sprachen wird die natürliche Intelligenz des Hörers/Lesers die intendierte Bedeutung entweder unter Einbeziehung von Kontext oder Weltwissen² selbst ermitteln oder aber durch Rückfragen klären.

Es ist wichtig zu bemerken: obwohl Fachsprachen oft - vor allem in ihren fachsprachlichen Anteilen - im Gebrauch strenger sind als natürliche (Umgangs)-Sprachen, so erben sie doch die Charakteristika der natürlichen Sprachen auf denen sie basieren.

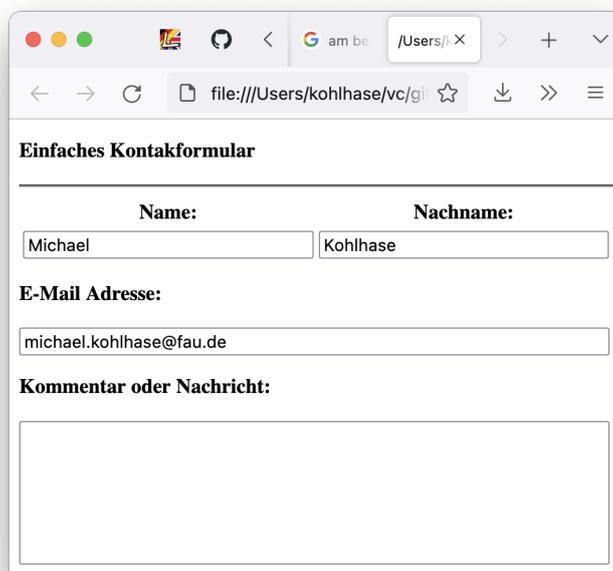
Allerdings gilt die hier beschriebene Dichotomie zwischen natürlichen und formalen Sprachen nicht uneingeschränkt. Hier ist die Digitalisierung ein starker Motor für die Entwicklung von Zwischenkonzepten.

macht, und dass Alexa die Bedeutung auch in allen Kontexten errechnen können muss; in diesem Beispiel muss dafür aber Information über den Ausgangsbahnhof aus dem physischen- oder Diskurs-Kontext erschlossen werden. Konsequenterweise wird Alexa zwar zur Licht-Steuerung in der Wohnung aber nicht zur Steuerung eines Atomkraftwerks eingesetzt.

²Unter **Weltwissen** verstehen wir das typische Allgemeinwissen der Gesprächsteilnehmer. Im Beispiel oben ergibt sich die Bedeutung „Sitzmöbel“ schon aus der Bedeutung des Verbs „setzen“ im Satz: Wie wir alle wissen kann man sich auf ein Sitzmöbel setzen, nicht aber auf ein Finanzinstitut. Eine metaphorische Verwendung im Sinne eines „Finanzinstituts“ ist zwar denkbar z.B. als „Peter setzt auf die Bank.“ aber sehr unwahrscheinlich.

3 Teil-Strukturierte Dokumente

Der erste wichtige Aspekt ist die Abhängigkeit der Strukturiertheit eines Dokuments wie oben beschrieben von den intendierten Verarbeitungsprozessen. So braucht man - je nach zu unterstützendem Prozess - keine vollständig formale Sprache. Ein gutes Beispiel sind (ausgefüllte) Formulare, wie z.B. das Kontaktformular einer Webseite in Abb. 2. Hier sind einige Teile formal (die Namensbestandteile und e-mail Adresse) und andere (der Nachrichtentext) natürlichsprachlich. Die formalen Teile sind ausreichend für eine Automatisierung der unmittelbar folgenden Arbeitsschritte: Übermittlung der Nachricht und Erstellung der Quittungs-EMail „Lieber Herr Michael Kohlhase, . . .“. Die Beantwortung der natürlichsprachlichen Anfrage selbst geschieht durch einen Menschen.



The image shows a screenshot of a web browser window displaying a contact form. The browser's address bar shows a local file path: file:///Users/kohlhase/vc/gi. The form is titled "Einfaches Kontaktformular" and is divided into several sections. The first section is for the name, with two input fields: "Name:" containing "Michael" and "Nachname:" containing "Kohlhase". The second section is for the email address, with an input field containing "michael.kohlhase@fau.de". The third section is for a comment or message, with a large, empty text area. The browser's interface includes standard navigation buttons and a search bar.

Abbildung 2: Kontaktformular

Das Kontaktformular aus Abb. 2 ist also hinsichtlich des Nachrichtentransports durchaus strukturiert - immerhin kann die Nachricht dem Kundenservice zugestellt werden und eine Antwort automatisch adressiert werden - aber nicht hinsichtlich der Beantwortung; dafür müsste die Antwortnachricht automatisch erstellt werden. Zugegeben, die Strukturierung und mögliche Automatisierung sind bei diesem Dokument recht gering.

Ein weiteres Beispiel für teil-strukturierte Dokumente ist die wohlbekannte Praktik, in normativen Dokumenten Paragraphen, Abschnitte und Sätze systematisch zu nummerieren, und diese dann anhand der Nummern zu referenzieren. Diese Praktik ist so systematisch, dass sie von Computerprogrammen interpretiert werden kann. Juristische Dokumente sind daher für die Referenzierung normativer Textfragment als strukturiert anzusehen.

Am anderen Ende der Strukturiertheitskala liegen die ausgefüllten Steuererklärungsformulare - als Papier oder in ELSTER. Diese sind hinsichtlich der Steuerfestsetzung als strukturiert zu sehen: das Steuerfestsetzungsprogramm der Finanzverwaltung kann 10% der erklärten Steuerfälle - die sogenannten „Autofälle“ - vollautomatisch bearbeiten. Die restlichen Fälle werden größtenteils nur deswegen an menschliche Experten ausgesteuert, weil die Risikomanagementkomponente falsche Eingaben vermutet. Details in [AB21].

4 Das Spannungsfeld zwischen Aufwand und Ertrag

Die oben genannten Beispiele zeigen bereits, dass die Strukturierung von Dokumenten und die Formalisierung von Wissen zusätzlichen Aufwand bedeuten. Eine Investition, die sich durch die dadurch ermöglichten Mehrwertdienste allerdings wieder amortisieren kann. Insgesamt wird sich die Formalisierung nur dann durchsetzen, wenn der Ertrag größer ist als die Investition. Abbildung 3 zeigt die Gewinnschwelle für einige bekannte Technologien der Informatik. Wir weisen darauf hin, dass die Diagramme in diesem Artikel alle nur qualitativ zu lesen sind. Aussagen wie „Die Investition für Formale Methoden sind viermal so hoch wie die für das Web 2.0“ oder „strukturierte juristische Dokumente lohnen sich im Schnitt“ sind nicht intendiert, obwohl sie strenggenommen aus dem Diagramm herausgelesen werden könnten, wenn man sie quantitativ läse.

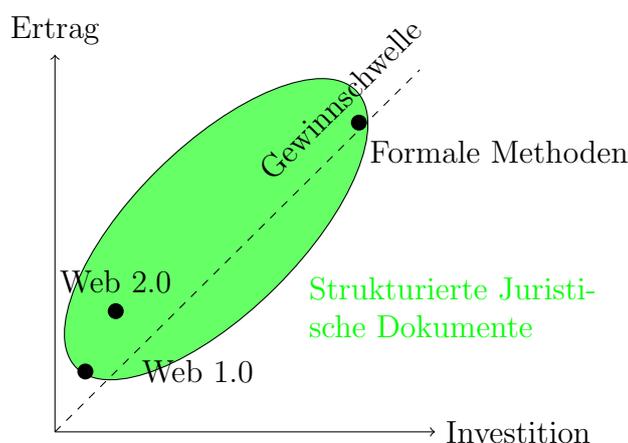


Abbildung 3: Wann lohnt sich Strukturierung?

Im World-Wide-Web (Web1.0) müssen strukturelle Elemente in Dokumenten durch spezielle Annotationen explizit gemacht werden. Durch das (formale) Adressierungssystem (URLs) des WWW können Referenzen auf Textfragmente formalisiert werden, und im Browser direkt abgerufen und angezeigt werden - damit wird das WWW zu einem großen Hypertext. Dass dieses System von strukturierten Dokumente die Gewinnschwelle überschreitet sehen wir schon an den geschätzten 1,7 Milliarden Dokumenten des WWW.

Beim Web 2.0, in dem Webseiten weiter zu interaktiven Web- Applikationen formalisiert werden, können Nutzer Texte, Bilder oder Filme bereitstellen und ermöglichen so Portale/-

Dienste wie Youtube, Facebook, oder GMail. Obwohl die Investition zur Programmierung solcher Portale wesentlich größer ist als beim Web 1.0 scheint der Nutzen der Technologie so riesig zu sein, dass das Web 2.0 boomt. Wir bemerken, dass Programmierung auch eine Form der Strukturierung ist: hier werden Aktionen und Prozesse formalisiert und so der Abarbeitung durch den Computer zugänglich gemacht.

Ein extremes Beispiel für dieses Phänomen sind die sogenannten „Formalen Methoden“ in der Informatik, wo - z.B. in der Programmverifikation die Bedeutung von Programmiersprachen so wie das intendierte Verhalten eines Programms in speziellen Logiksprachen spezifiziert werden. In solchen Logiken lassen sich sogar die Schlussweisen in (mathematischen) Korrektheitsbeweisen mittels spezieller Computerprogramme - sogenannten Beweisprüfern - nachvollziehen, und so die vollständige Abwesenheit von Programmierfehlern sicherstellen. Hierbei ist allerdings die notwendige Investition in Spezifikation und Erstellung des maschinell verifizierbaren Korrektheitsbeweises so riesig - die Kosten der Spezifikation und erst Recht einer Verifikation übersteigen die Kosten des ursprünglichen Programms bei weitem - dass von dieser Möglichkeit nur bei sicherheitskritischen Programmen Gebrauch gemacht wird. Dann allerdings können auch riesige Schäden vermieden werden (so hat Intel 1994 durch den bekannten FDIV Fehler im Pentium Chip einen Verlust von ca. 475.000.000\$ erlitten und verifiziert seitdem alle Chips und systemnahe Software mit Formalen Methoden; siehe [Pen]). Die Formalen Methoden überschreiten in solchen Fällen die Gewinnschwelle, allerdings nicht so deutlich wie bei den Web-Technologien, Formale Methoden sind (immer noch) eine Nischentechnologie für Hochsicherheitsanwendungen.

Der grüne Bereich in Abb. 3 versucht das Potential von strukturierten Dokumenten im Zivilprozess abzuschätzen. Dieses reicht von der digitalen Dokumentenverwaltung in einem „Prozess-Hyperdokument“ bis zu einer logischen Modellierung der juristischen Subsumption und dadurch theoretisch denkbaren automatischen Verifikation oder sogar vollautomatischen Erstellung von Urteilen und deren Begründungen (siehe unten). In Abb. 3 liegt der grüne Potentialbereich nur teilweise oberhalb der „Gewinnschwelle“. In der Tat werden nicht alle möglichen juristischen Dienste die dafür erforderliche Strukturierungsinvestitionen wieder einspielen. Aber wie die Abbildung schon andeutet, scheint es plausibel, dass es durchaus eine große Klasse von Anwendungen für strukturierte juristische Dokumente und darauf basierende Dienste gibt, die Produktivität der Justiz in Zivilprozessen signifikant erhöhen kann.

4.1 Produktivitätsprofile für strukturierte Dokumente

Im folgenden wollen wir die Kriterien untersuchen, nach denen Elemente der Dokumentenstrukturierung und die darauf aufbauenden Dienste bewertet werden können. Wir wollen dazu neun paradigmatische Fälle genauer betrachten. Wie wir bereits gesehen haben, ist es vorteilhaft, Strukturierung/Formalisierung nicht als eine absolute Eigenschaft zu sehen, sondern als graduell/flexibel³: in einem Dokument können verschiedene Aspekte verschie-

³[Koh13] diskutiert flexible Formalisierung mathematischer Dokumente. Trotz der vordergründig anderen Domäne ist diese Diskussion hier relevant, da Mathematik und Jurisprudenz beid

den weit formalisiert werden, und andere informell belassen werden. In Abb. 4 tragen wir den Formalisierungsgrad gegen die dadurch erreichbare Funktionalität von verschiedenen (zum Teil noch hypothetischen) Diensten ab. Hierbei ist zu beachten, dass im Gegensatz zu Abb. 3, nicht die Gesamtinvestition gegen den Gesamtertrag abgetragen wird, sondern das Verhältnis der Formalisierungs-Breite/Tiefe gegen das Verhältnis des damit realisierbaren Ertrags zum möglichen Gesamtertrag. Intuitiv: Abb. 4 zeigt wie schnell die einzelnen Mehrwertdienste auf die Formalisierung ansprechen.

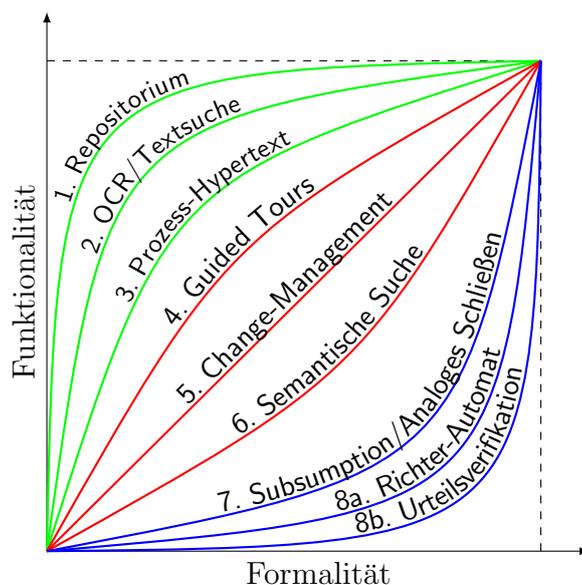


Abbildung 4: Dienste und Strukturiertheit

1. Repository Auf der untersten Ebene der Digitalisierung von Dokumenten ist die elektronische Bereitstellung aller Dokumente auf einem Dokumentenverwaltungssystem (Repository). Wichtig: In einem solchen Repository haben alle Dokumente und ihre Versionen eine eindeutige „Adresse“ mittels derer auf sie zugegriffen werden kann. Dadurch können die jeweils berechtigten Mitarbeiter ortsunabhängig und parallel auf die jeweils neueste Version der Dokumente zugreifen. Viele Repositorien gewähren auch Zugang zu vorherigen Dokumentenversionen und Entwicklungshistorien. Die „Dokumenten-Adresse“ (ggf. mit Versions-Information und Metadaten) ist die einzige formale Anteil auf Repositoriums-Ebene.

2. OCR/Texterkennung Bisher haben wir noch keine Annahmen über die Repräsentation der Dokumente selbst gemacht, es könnte sich hierbei also durchaus um gescannte Handschriften handeln. In einem weiteren Formalisierungsschritt könnte also durch optische

Argumentations/Beweis-orientiert arbeiten, und in der Argumentation eine sehr hohe Präzision voraussetzen, die - zumindest momentan - nur durch Formale Methoden in irgendeiner Form geliefert werden kann

Zeichen/Schrifterkennung (OCR) digitale Texte entstehen, die eine Textsuche möglich machen. Dabei ist wichtig darauf hinzuweisen, dass, obwohl die OCR automatisch abläuft, es sich hier um einen Formalisierungs/Strukturierungsschritt handelt, der in der Regel fehlerbehaftet ist und im Extremfall durch manuelle Arbeit korrigiert und vervollständigt werden muss. Allerdings (und das zeigt der Kurvenverlauf in Abb. 4) führen Qualitätsverbesserungen in der Texterkennung zu besserer Textsuche. Wenn bei der Textsuche Vollständigkeit kritisch ist, z.B. um sicherzustellen, dass ein bestimmter Sachverhalt unerwähnt bleibt, kann sich der Mehraufwand der „Formalisierung durch manuelle Korrektur“ durchaus lohnen.

Wenn wir die Kurven „1. Repository“ und „2. OCR/Textsuche“ in Abb. 3 vergleichen sehen wir, dass die erstere oberhalb der zweiten verläuft. Das heißt, dass bei Formalisierung für das Repository bereits mit sehr wenig Formalisierungsaufwand eine hohe Wirkung erzielt werden kann. Im Vergleich dazu muss für die Textsuche noch zusätzlich formalisiert werden. Mehrwertdienste, die eine konvexe Kurve haben (bauchig nach oben) sind attraktiv, weil sie sich einfach realisieren lassen. Entsprechend finden wir diese Technologien auch tendentiell unten links im Diagramm in Abb. 3.

3. Prozess-Hypertext In einem weiteren Formalisierungsschritt könnten alle sinntragenden Dokumentenfragmente gekennzeichnet und mit eindeutigen Kennungen versehen und alle Querverweise auf solche anhand dieser Kennung explizit gemacht werden. Damit bekommen wir einen Prozess-Hypertext, die in einem (hypothetischen) Prozessbrowser das manuelle Blättern in verschiedenen Papier-Akten unnötig macht.

Damit haben wir die Gruppe der weitgehend syntaktischen Strukturierungen/Dienste in Abb. 4 erschöpft. Die nächste Gruppe in rot benötigt „semantische Formalisierungen“; hierunter verstehen wir eine formale/strukturierte Charakterisierung der Art der Querverweise. Im Prozess-Hypertext (und auf dem World-Wide-Web) kann ein Querverweis von A nach B verschiedenste Relationen bedeuten: z.B.

- **Siehe auch:** Um A besser zu verstehen ist es auch interessant sich B anzugucken.
- **Definition:** A ist definiert durch/in B .
- **Empfehlung:** Personen, die A angeguckt haben, haben schließlich B gekauft.
- **Kontrast:** A ist das Gegenteil von B (in einem geeigneten Kontext)

Offensichtlich ermöglicht eine nähere Spezifikation der Relation weitere Mehrwertdienste; Menschen können die Natur des Querverweises meist aus dem Kontext von A erschliessen und darauf adäquat reagieren.

4. Guided Tours Eine Formalisierung der Abhängigkeit zwischen sinntragenden Textfragmenten (Fragment A ist von Fragment B abhängig, wenn man B kennen muss um A voll zu verstehen) kann genutzt werden, um semantisch abgeschlossene Dokumente zu erzeugen - wir nennen diese auch Guided Tours; siehe [Koh+11]. Ein Prozessbeteiligter kann sich für jedes sinntragende Textfragment eine Guided Tour erzeugen lassen, das dem Leser alles nötige Vorwissen in der fachlich richtigen Reihenfolge darbietet. Damit entfällt nicht nur das Blättern nach im Text gegebenen Querverweise, sondern es werden auch die nicht im Text gegebenen Abhängigkeiten im Fallwissen bereitgestellt. Erweitern wir den Dienst noch um ein „Gedächtnis“, was der Leser bereits gesehen hat, so können wir auch

die Nutzeradaptive Duplikation bereits bekannter Sachverhalte vermeiden.

5. Änderungsmanagement Die Umkehrung der Abhängigkeitsrelation kann dann zusätzlich zum Änderungsmanagement verwendet werden: Hängt A von B ab, und B ändert sich, so müssen wir A noch einmal prüfen, ob die Änderung von B eine Änderung von A notwendig macht. Annotieren wir zusätzlich zur Abhängigkeitsrelation auf Textfragmenten noch die Abhängigkeit von Werten (z.B. Zahlenwerten wie z.B. die Höhe des Streitwertes), so kann die Propagierung geänderter Werte in andere Dokumente (teil-)automatisiert werden. Dies könnte bei einer Textbaustein-basierten Verwaltung der Sachvorträge der Gerichte (z.B. wie im in [Ziv21] vorgeschlagenen gemeinsamen Basisdokument) eine große Rolle spielen.

6. Semantische Suche Für eine semantische Suche im Prozess-Hypertext, um z.B. Fragen zu klären wie „*Mit welcher Mehrheit wurde der Vorstand in der letzten Hauptversammlung entlastet?*“ brauchen wir noch weitergehende Formalisierung des Texts: wir können nicht einfach auf eine Textsuche vertrauen, da die „*Hauptversammlung*“ in den Dokumenten auch als „*Aktionärsversammlung*“ bezeichnet werden könnte (Synonyme) oder das Abstimmungsergebnis statt einer (expliziten) Prozentzahl durch Begriffe wie „ *einstimmig*“ oder „*mit grosser/qualifizierter Mehrheit*“ (Umschreibung) verbalisiert werden könnte (Schlüsse aus dem Weltwissen über Abstimmungen). Eine semantische Suche bezieht Relationen wie Synonyme, Hyper/Hyponyme (Spezialisierung/Verallgemeinerung), und Umschreibungen mit ein und liefert so ein umfassenderes Ergebnis als reine Wortvorkommenssuche, wie sie traditionelle Suchmaschinen anbieten.

Die letzte Gruppe von Diensten (in Abb. 4 blau dargestellt) benötigt eine noch (sehr viel) höhere Formalisierung, so dass juristische Schlussweisen explizit gemacht werden oder sogar automatisiert werden können.

7. Juristische Schlussweisen Dokumente im Zivilprozess enthalten ausser Tatsachenfeststellungen auch juristische Argumentationen und Begründungen, die sich ihrerseits auf Schlussweisen wie

- **Subsumption**, d.h. Anwendung einer juristischen Norm auf einen konkreten Sachverhalt; siehe auch [AKR21],
- **Priorisierung** zwischen mehrererlei möglichen juristischen Argumentationen oder Begründungen oder
- **analoges Schliessen**, d.h. Ausnutzen einer Gleichartigkeit zur Übertragung von Rechtskonsequenzen; siehe auch [KAR21]

stützen. Dabei handelt es sich aus der Dokumentenperspektive um komplexe Relationen zwischen Textfragmenten. Zu deren Explizierung können diese entweder manuell annotiert werden oder aus einer genaueren Analyse der Textfragmente - und vor allem dem darin beschriebenen Fakten, dem juristischen Fachwissen und dem allgemeinen Hintergrundwissen erschlossen werden (siehe unten). Für diese „Wissensperspektive“ müssen wir allerdings die Granularität der Modelleriung verfeinert von der Textfragmentenebene auf die Wort-Ebene, da diese die atomaren Wissensträger sind. Dieses bedeutet natürlich eine viel Größere Basisinvestition, bevor Dienste auf diese Relationen aufbauen können, so dass die Kurve in Abb. 4 konkav (bauchig nach unten) dargestellt ist.

8. Richter-Automat & Urteilsverifikation Bei der hier postulierten „*Urteilsverifikation*“ handelt es sich um einen Dienst wie der oben beschriebenen Programmverifikation, in dem Urteile so genau formalisiert werden, dass die zugrundeliegenden Argumentation, Referenzen und Subsumptionsbeziehungen vom Computer geprüft werden können. Dadurch könnten Fehlschlüsse oder Argumentationslücken aufgedeckt werden oder das Vertrauen in deren Gültigkeit gestärkt werden und so Revisionsverfahren minimiert werden. Hierbei müssen nicht nur die Prozessdokumente selbst formalisiert werden, sondern auch die normativen Dokumente (BGB, StVO, etc) sowie das fachliche Hintergrundwissen (im wesentlichen der in einem Jurastudium vermittelte Wissensschatz) und sogar das Allgemeinwissen (z.B. dass PKWs Kraftfahrzeuge sind), also all dem Wissen das einem Urteil oder Vergleich zugrunde liegt.

Der hier postulierte „*Richter-Automat*“ (vergleiche auch [Adr17]) geht noch eine Stufe weiter, hier werden Urteile durch den Computer automatisch erstellt⁴.

In Abb. 4 landet die Kurve für den Richterautomat paradoxerweise noch vor der Urteilsverifikation, da die erstere weniger Formalisierung (des Urteils selbst) erfordert beim gleichen Ertrag (ein formal verifiziertes Urteil). In der Praxis wird es - wenn diese Dienste wirklich realisierbar sein sollten - wie bereits für die Programmverifikation eine Mischung aus menschlicher Anleitung und maschineller Argumentations-Automatisierung in einer integrierten „*Zivilprozessentwicklungsumgebung*“ hinauslaufen, die alle hier beschriebenen Mehrwertdienste integriert.

Zusammenfassung Diese Diskussion gruppiert den Raum der möglichen Mehrwertdienste anders als die LegalTech Community, die (siehe [Leg] oder [Goo15]), die ihre Technologie in Anlehnung an die Phasen industrieller Disruptionen in drei „Phasen“ einteilen. Dort wird unter dem Stichwort

1. „LegalTech 1.0“ vor allem Büroautomatisierung und Kommunikationssoftware, unter
2. „LegalTech 2.0“ begrenzte, automatisierte Rechtsdienstleistungen und
3. „LegalTech 3.0“ Technologien verstanden, die das Berufsbild menschlicher Anwälte und Richter grundlegend zu verändern.

Diese Unterscheidung ist rein phänomenologisch, die Klassifikation der Mehrwertdienste in diesem Artikel orientiert sich an Kosten-Nutzen Verhältnissen bei der Formalisierung/Strukturierung.

5 Fazit & Ausblick

Motiviert durch den Ruf nach „strukturierten Prozessakten“ im Diskussionspapier der Arbeitsgruppe „Modernisierung des Zivilprozesses“ [Ziv21] im Auftrag der Präsidentinnen und Präsidenten der Oberlandesgerichte, des Kammergerichts, des Bayerischen Obersten

⁴Für diesen Artikel tun wir so, als ob die Erstellung von Urteilen und Vergleichen die Hauptaufgabe von Richtern ist und vernachlässigen wichtige Aufgaben wie die Tatsachenfeststellung und Verhandlungsführung, die weniger auf Dokumenten beruhen, und sich möglicherweise noch schwieriger formalisieren lassen.

Landesgerichts und des Bundesgerichtshofs und den dadurch losgetretenen Diskussionen analysiert dieser Artikel den Begriff des „strukturierten/formalen Dokuments“ aus der Sicht der Informatik und der Künstlichen Intelligenz.

Aus der Basisfeststellung: „*Ein Dokument ist strukturiert, wenn auf dessen formalen Elementen geeignete Mehrwertdienste aufbauen können*“ entwickeln wir eine Einordnung potentieller Digitalisierungsdienste für den Zivilprozess und der dafür jeweils notwendigen Formalisierungsinvestitionen. Diese Dienste reichen von einfacher Automatisierung des Dokumentenmanagements, das heute schon realisiert werden sollte, bis hin zu einem umfassenden „Richterautomat“, der in der hier suggerierten Allgemeinheit wahrscheinlich noch weit in der Zukunft liegt, da wir hier Dokumente sehr tiefgehend formalisiert und sehr weitgehende weitere Wissensquellen formal erschlossen werden müssen. Für sehr eingeschränkte Bereiche hingegen scheint eine Automatisierung juristischer Verarbeitungsprozesse möglich, wie das Beispiel der Erstellung von Klageschriften bei Verletzungen von Fluggastrechten bei Flightright [Fli] oder der automatischen Steuerfestsetzung [AB21] schon zeigen.

Der vorliegende Artikel wollte ein Bewusstsein dafür wecken, die Egtenschaft „formal/strukturiert“ nicht als absolut gesehen werden kann: Die Formalisierungs-Tiefe/Dichte einen großen Einfluss auf die späteren Mehrwertdienste. Je „intelligenter“ die Dienste sein sollen, so mehr Aufwand muss in die Formalisierung/Strukturierung der Dokumente investiert werden. Insbesondere ist es ein Denkfehler zu glauben, dass mit einer Formalisierung/Strukturierung eines Aspektes („ein-für-allemal“) alle Anwendungen unterstützt werden können.

Es scheint klar, dass die zur Digitalisierung notwendigen Entwicklungen nur gemeinsam von Rechtswissenschaften und Informatik durchgeführt werden können. Dabei wird eine sehr wichtige Voraussetzung sein, zwischen Informatik und Rechtswissenschaften eine gemeinsame Sprache und ein Verständnis für die jeweils anderen Anliegen und Vorgehensweisen zu entwickeln. Die dafür nötige Forschung und Entwicklung kann in kollaborativen Projekten durchgeführt werden.

Literatur

- [AB21] Axel Adrian und Holger Barthel. “Expertensysteme im Bereich der Steuerverwaltung als Beispiel eines strukturierten digitalen Verfahrens – Ein Erfahrungsaustausch zum Nutzen eines künftigen digitalen Justizportals”. Im Tagungsband *Digitalisierung von Zivilprozess und Rechtsdurchsetzung* in Vorbereitung. 2021.
- [Adr17] Axel Adrian. “Der Richterautomat ist möglich – Semantik ist nur eine Illusion”. In: *Rechtstheorie* 48.1 (2017), S. 77–121.
- [AKR21] Axel Adrian, Michael Kohlhase und Max Rapp. “A novel understanding of legal syllogism as a starting point for better legal symbolic AI systems”. In: *24. Internationalen Rechtsinformatik Symposium (IRIS 2021)*. Hrsg. von Erich Schweighofer und Walter Hötzendorfer und Franz Kummer und Ahti Saarenpää

- und Stefan Eder und Philip Hanke. 2021, 169ff. URL: https://jusletter-it.weblaw.ch/issues/2021/27-Mai-2021/05_rechtstheorie_01._863f4db39a.html__ONCE&login=false.
- [Fli] *Flightright: Ihr Experte für Ticketerstattung und Flugentschädigung*. URL: <https://flightright.com> (besucht am 10.10.2021).
- [Goo15] Oliver R. Goodenough. *Legal Technology 3.0*. Blog post. 2015. URL: https://www.huffpost.com/entry/legal-technology-30_b_6603658.
- [KAR21] Michael Kohlhase, Axel Adrian und Max Rapp. “Context Graphs for Ampliative Analogical Legal Reasoning and Argumentation”. In: *24. Internationalen Rechtsinformatik Symposium (IRIS 2021)*. Hrsg. von Erich Schweighofer und Walter Hötzendorfer und Franz Kummer und Ahti Saarenpää und Stefan Eder und Philip Hanke. 2021, 231ff. URL: https://jusletter-it.weblaw.ch/issues/2021/27-Mai-2021/05_rechtstheorie_03._3ed8681731.html__ONCE&login=false.
- [Koh+11] Michael Kohlhase u. a. “The Planetary System: Web 3.0 & Active Documents for STEM”. In: *Procedia Computer Science 4* (2011): *Special issue: Proceedings of the International Conference on Computational Science (ICCS)*. Hrsg. von Mitsuhsa Sato u. a. Finalist at the Executable Paper Grand Challenge, S. 598–607. DOI: 10.1016/j.procs.2011.04.063.
- [Koh13] Michael Kohlhase. “The Flexiformalist Manifesto”. In: *14th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012)*. Hrsg. von Andrei Voronkov u. a. Timisoara, Romania: IEEE Press, 2013, S. 30–36. ISBN: 978-1-4673-5026-6. URL: <https://kwarc.info/kohlhase/papers/synasc13.pdf>.
- [Leg] *Legal Technology*. Wikipedia. URL: https://de.wikipedia.org/wiki/Legal_Technology (besucht am 01.11.2021).
- [Pen] *Pentium-FDIV-Bug*. Wikipedia. URL: <https://de.wikipedia.org/wiki/Pentium-FDIV-Bug> (besucht am 06.10.2021).
- [PH21] Erich Schweighofer und Walter Hötzendorfer und Franz Kummer und Ahti Saarenpää und Stefan Eder und Philip Hanke, Hrsg. *24. Internationalen Rechtsinformatik Symposium (IRIS 2021)*. 2021.
- [Rum+] Bernhard Rumpe u. a. *Digitalisierung der Gesetzgebung zur Steigerung der Digitalen Souveränität des Staates*. Techn. Ber. 19. Nationales E-Government Kompetenzzentrum. URL: <https://negz.org/wp-content/uploads/2021/06/NEGZ-Kurzstudie-19-Digitalisierung-der-Gesetzgebung-2021.pdf>.

[Ziv21] Arbeitsgruppe „Modernisierung des Zivilprozesses“. *Modernisierung des Zivilprozesses – Diskussionspapier*. Im Auftrag der Präsidentinnen und Präsidenten der Oberlandesgerichte, des Kammergerichts, des Bayerischen Obersten Landesgerichts und des Bundesgerichtshofs. 2021. URL: https://www.justiz.bayern.de/media/images/behoerden-und-gerichte/oberlandesgerichte/nuernberg/diskussionspapier_ag_modernisierung.pdf.