Towards a Flexiformal Legal Domain Model (FlexiLDM)

Osman Anil Basaran [1] and Michael Kohlhase [2]

Abstract: We propose a flexiformal approach to building a Legal Domain Model as a precondition for symbolic and neuro-symbolic legal reasoning (the "Flexiformal Legal Domain Model", abbr. "FlexiLDM“). It proposes sTeX and other document formats as the intermediate representation language inside the FLAMS-framework to combine human-oriented legal prose with machine-processable structure. The FlexiLDM containts the knowledge that legal pracitions have learned during their training and that is necessary to perform legal reasoning. The specific tasks has usually never been written down suffiently enough. Flat hierarchies and flat semantics inside swarm ontologies shall enable developers to create interactive public legal services to assess citizens in enforcing their constitutional rights on access to justice.

Keywords: Knowledge Representation, Ontology, Semantics, Flexiformal, Legal Informatics, Legal Tech, Legal Services, Access to Justice, Commercial Register

1 Introduction

There is a new buzzword in town: "Law as Code"! A rather sensational headline in sheep's clothing which led the German Federal Agency for Disruptive Innovation (SPRIN-D) to spend millions of Euros on an encoded legislation process [Ru25]. The underlying idea is that if we can make law a bit more like code, then we could inherit some of the advantages that code – i. e., direct execution, testing, formal verification, and change management – may carry over and make the law more directly applicable, resilient, and manageable.

The underlying technology is one of *formalization*: (program) code is a formal, i. e., machine-actionable representation of procedural knowledge, in which the important aspects are made so explicit, that it can be directly executed on a computer. There are multiple forms of formalization. Formalization into program code is one; it is suited for direct execution of formalized procedures. Formalization into logical systems is another option; it supports the mechanization of reasoning and argumentation. A third is formalization into knowledge representation formats and ontologies; that supports the automation of taxonomic and relational reasoning.

Generally, formalization eliminates ambiguity, vagueness, and elision of detail at the cost of diminished flexibility, scope of application, and legibility. The ultimate benefit is that formalized representation – e. g., of the law – can be the basis for support services. An

1 FAU Erlangen-Nürnberg, Computer Science, Martensstr. 3, 91058 Erlangen, osman.anil.basaran@fau.de, https://orcid.org/0009-0008-6700-4264
2 FAU Erlangen-Nürnberg, Computer Science, Martensstr. 3, 91058 Erlangen, michael.kohlhase@fau.de, https://orcid.org/0000-0002-9859-6337

example of such services is the partial automation of the evaluation of tax filings based on a formalization of rules; see [Ko22] for details. But there are times when one must not, should not, cannot or does not want to accept diminished flexibility, scope of application, and legibility, but would still like to embrace law's formalization with all of its benefits.

Research Gap Most modern Legal Knowledge Representation systems such as the DAPRECO project [Ro20], LegalRuleML [Pa11] and the Akoma-Ntoso Knowledge-Base [FLL22] follow the "Rule-as-Code" and "Code-as-Law" approach [Mo20; Mo25a; Mo25b] towards isomorphism [CB91]. These approaches may bridge the gap between knowledge representation and computational reasoning to aid legal practitioners in their primary activities by providing simple tools for workflow modeling and explanations [Ru25; Ra25; FI26]. But they often fall short in other areas: most input and output data/text is difficult to read and comprehend for regular end-users as data tends to be inaccessible. Consequently, developers have to find ways to transform glorified knowledge-graphs into individually tailored, human-comprehensible, "laymen-understandable" documents. Changes to knowledge bases or target documents cannot be directly authored inside them. This clearly hinders effective scalability of legal tooling. The question is: why not embed both the knowledge and the explanatory documents into a single unified format?

Contribution Some of these problems in the legal domain are something which have already been overcome in the mathematical domain through *flexiformalizing* – formalizing to a flexible level of formality – domain knowledge and embedding them as semantic annotations into document sources, which can be transformed into (inter-) active and reader-adaptive documents via a background knowledge management server; see [Ko13; Mü26] for details. The contribution of this paper is to make this flexiformalization ecosystem (initially) developed for mathematical knowledge available for the legal profession that is much less technically oriented. It will satisfy both the technicians' and the public's needs, i. e., the needs of the ordinary people and the public institutions, incl. the legal practitioners themselves: technicians will be able to harvest formal knowledge from and by legal practitioners themselves. The public will be offered new open-sourced (!), interactive public services by the very same technicians. One prime example is the offering of easy-to-understand explanations and validations of legal documents. The first part falls under the category "Access to Justice" [CG78; CGJ77].

Overview In Sect. 2 and Sect. 3, we will review the state of the art. Building on that we present the main case study for flexiformal legal code in Sect. 4. Sect. 5 concludes the paper and discusses future work.

Acknowledgements The work reported here was funded by the National and the Bavarian Chamber of Notaries and the Bavarian Ministry of Justice in the DIREGA research project.

2 From “Code is Law” to “Rules as Code”

Code is Law and Code as Law The term “Code is Law” describes how code shapes and constrains behavior. This process is functionally comparable to legal rules. In the past, this would be regarded as a threat to citizens’ liberty [Ea96; Le99; Le00]. The minor evolution of “Code as Law” followed soon with the rise of Smart Contracts (self executing contracts) [Sz94; Zh20]: Smart Contracts are based on the distributed ledger technology, where legal mechanisms – e. g., transactions and dispute resolutions – are enforced peer-to-peer without (proper) regard to lawfulness and without any intervention of intermediate institutions or the peers themselves [Na08; Jü21; Qu22].

Law as Code The term “Law is Code” turned into “Law as Code” (LaC) and describes from here on the progressive (reciprocal) transformation of the law into terminating code. This includes all work results from the fields of legal informatics and legal tech such as legal knowledge representation, formalization, logic and reasoning [FH18; Os21].

Rule as Code After the Smart Contract hype [ZRH21] and the exhaustively laid theoretical foundations in computational law and legal reasoning, we have finally entered a productive stage of the “Rule as Code” approach (RaC) [Ne18; Go20; Wa21]. The mission of RaC is to support, enhance and complement the legislation process and public services by drafting law texts, which are human-readable and machine-interpretable by design, further by turning existing legislation into machine-readable code by guaranteeing “Compliance by Design“ [Mo19; MR20b; MR20a; Li24].

Opportunities of LaC Compared to other approaches, LaC[3] is of course recognized to bear lots of chances and advantages:

1. acceleration of/efficiency gains in the legal drafting process;
2. machine-interpretability of the encoded machine-validational law;
3. accurate and deterministic machine-interpretation and reasoning in comparison to generative pre-trained transformer-models;
4. transparency, trustfulness, faithfulness, and clarity of the encoded law;
5. availability of many open-sourced solutions;
6. growth in synergy effects between the different sectors by reducing the need for deep multidisciplinary cooperation.

Challanges, Risks and drawbacks of LaC On the other hand, LaC bears its challenges, risks, and drawbacks:

[3] Anything related to LaC can be applied on RaC and the other way round.

1. over-complexity and over-formalization of the code when seeking low discrepancy between normativity and descriptiveness, and between the closed and open world;
2. ex-ante-formalization and ex-post-enforcement with only few regards to time;
3. hard accessibility, logistics, and maintenance of the code base by computer scientists;
4. sharing huge knowledge bases in an interoperable linked format;
5. loss of flexibility/stiffness/unintended gaps in granular formalization models and information retrieval, e. g., regarding constitutional rights interpretation methods, value-based principles, open-textured rules, and context;
6. undermining authority of courts and ignoring unforeseen situations of the law when interpreting legal text (governance by computer scientists);
7. unclear shifts in responsibility, interpretation, and liability reaching from public to private partners;
8. consideration of legislation, public services et al. as high-risk use cases with high-compliance-requirements, e. g., according to the EU AI-Act;
9. barely any development towards at least low-level human-readable and comprehensible documents through user-centric Legal Design [KJS24]/user-interfaces aid citizens in taking advantage of public services (Access to Justice by Design as stated in Sect. 1).

Especially on the comprehensiveness- and accessibility-side of things there is a lot of room for scientific and practical contribution.

3 Flexiformal Knowledge Representation and Active Documents

The legal domain is not the only domain, that has to deal with such challenges and risks. The domain of mathematics has done so for past decades. The only difference: the involved stakes are much lower than in the legal domain or even some of the computer science. In high stake fields of computer science, such as high-risk system validation and out-of-order-theorem-proving with regard to cybersecurity, mistakes can lead to tremendous liability cases. Mistakes in logic and formalization cannot be pardoned.

The counterpart of LaC in mathematics could be seen as "Math as Code" and has been researched for more than a century. After developing the "rigorous" mathematical style of communication, the foundational crisis of mathematics in the late 19th century led to Hilbert's program, which called for systems, theoretical could formalize the foundations of mathematics and could moreover show the absence inconsistencies. While the second aim ultimately failed, formalized mathematics is thriving, but is plagued by the same problems the ones for LaC described above. To remedy this, the KWARC group at FAU Erlangen-Nürnberg has developed the notion of "flexiformalizing" [Ko13] – formalizing to flexible levels of formality and embedding the flexiformal annotations into mathematical typesetting formats like LaTeX. The main application for flexiformalization is to enable services that need less formalization than the verification of correctness of mathematical proofs, e. g., mathematical search engines and computer supported learning (in mathematics and computer science). This naturally leads to the notion of **active documents** [Lo23] – interactive electronic documents with embedded value-added services that adapt their contents, presentation, and available services to the competencies and prior knowledge of the reader to make communication efficient and effective.

3.1 FTML, FLAMS, and FTML Surface Languages

To make these ideas operational and evaluate them in practice, the KWARC group has developed the FTML (Flexiformal HTML) representation format – an extension of HTML by special attributes that represent flexiformal annotations – and the Flexiformal Annotation Management System (FLAMS). FLAMS harvests FTML documents, integrates them into a knowledge graph, provides advanced knowledge management services, and serves FTML documents instrumented with added value services to your browser; see [Mü26] for details. FLAMS has been used as the basis of the ALeA system (Adaptive Learning Assistant [Lo23; KW26a]) in 8 courses 8 semesters with up to 1000 students per semester; this shows that flexiformal representation and active documents are ready for academic and professional use.

For mathematics, physics and computer science, we can use the sTeX package to embed semantic annotations in LaTeX documents. The regular `pdflatex` workflow allows to generate regular PDF documents that hides the semantic annotations and the RusTeX system to generate FTML. For other disciplines that do not use LaTeX, there are the WOIDE [**{Adr25}**] and CPoint IDEs [Ko08] that allow semantic annotations in MS Word and PPT and the FloDown extension of MarkDown [Mü26] that can generate FTML for FLAMS and thus active documents.

The main idea of flexiformalization in FTML/FLAMS is that all concepts, objects, and relationships are represented as **symbols** which in turn are identified via special FTML URIs. In flexiformal documents, terminology can be annotated by symbols, definitions, statements of properties and relationships are annotated by their epistemic (and/or didactic) role. In particular the symbols support automation of communication and learning support services, especially, when coupled with reader competency modeling: If we have a good coverage of technical document with symbols for the terminology employed and formalizations for their relationships, FLAMS can automatically generate in-place definition and example lookups, explanations tailored to readers, and place diagnostic callbacks that in turn inform the reader models.

The investment into flexiformalization is offset by the improved interoperability, re-usability, and thus potential for sharing of document fragments. This is facilitated by a general/shared **domain model** – a modular collection of symbols, their definitions, and property assertions. For mathematics the Semantic Multilingual Glossary of Mathematics (SMGloM) fills this need, it consists of over 6.500 symbols with their definitions, some examples, and theorems organized in ca. 2.800 modules that modulate concept inheritance [Ko26b].

4 The Flexiformal Legal Domain Model

An analogous conceptual and representation infrastructure – combined with law-specific services – could be used for generating public service documents: for one without heavy tooling around the represented knowledge, and for two without relying too heavily on the RaC cult in a time when its goals, specifications, and features are still unclear. *Flexiformalizing* means embracing the opportunity, to formalize as little knowledge as needed, and as much as sufficient to enable proper reasoning through inference systems such as `Catala` [MCP21] and `sCASP` [Ar18; Va24; Ad25].

4.1 Overcoming the differences between Legal and Mathematics

Differences between Legal and Mathematics Despite all the similarities (Sect. 3), there are still some crucial differences between the legal domain and the mathematical domain. To call out some, these are particularly [Li23]:

1. ambiguity, vagueness, and openness of the "non-ordinary" legal language;
2. consideration of interpretational methods, principles, values, legal practices, such as principles of burden of proof in legal practice, and mechanisms for resolving conflicts on different argumentation levels;
3. variety of multi-level-authorities and legal systems on diverse granular regulation levels (supra-national, inter-national, national, (sub-) federal);
4. variety of addressees (legal, para-legal, laymen, clients, teachers/scholars, learners) of (authoritative) legal text, e. g., legal opinions, cases, statutes;
5. Cross-domain knowledge sourcing from different perspectives, e. g., legislation, judiciary, administration, research, practice, and informal dispute.

Transferring mathematical concepts to the legal domain Some of these differences are very hard to capture formally. But this can be overcome by simply intentionally skipping the intense and granular formalization of such AI-hard legal terms, like we already do in the mathematical domain. This is what we call flexiformalizing, cf. Sect. 1. This not only leaves visible room for the human-in-the-loop, but also allows for low-level reasoning and low-level interactive documents for the user.

Solving the cross-domain knowledge sourcing Either legal people like it or not: someone will somewhere sometime have to fill the knowledge base bottom-up with meaningful substance to finally create the (in our case) Flexiformal Legal Domain Model (FlexiLDM) [Ba26b] with the Flexiformal Ontology inside [Ko26a].[4] The only concern here could be,

[4] This certain someone should always be a human who can be supported with automation and annotation tools.

from *where* the knowledge shall be sourced. Without re-inventing the wheel, the knowledge shall be gathered from authoritative sources as suggested in the ANGELIC-II-paper [AB23], expanded by literature on the terminology of the law:

1. laws themselves;
2. official issue guidance, e. g., Guidelines [Eu26], Recommendations, and Best Practices;
3. legal commentaries, incl. interpretational aids;
4. legal glossaries, dictionaries, and encyclopedia, e. g., [Le26a];
5. decisions in leading cases, i. e., precedents and case digests;
6. practitioners' expert knowledge, incl. guidance leaflets for clients, guidance manuals, and checklists for staff.

4.2 Modelling in multiple languages and multiple jurisdictions

All of the methods above are ultimately based on the concept of a (mathematical or legal) *symbol*, i. e., a uniquely identifiable (mathematical or legal) concept with an associated global name (a URI for practicality). Note that the symbol is a *platonic notion*: the concept itself, independent of the words used to reference it in natural langauge. For instance, the abstract platonic concept of an automobile could be reference by the URI `https://mathhub.info?a=things,p=vehicles/motor,m=fourwheeler?automobile` and referred to by the words "car" and "automobile" in English, by "Auto" in German, and "voiture" in French. The words that are used to refer to symbols in natural languages are called *verbalizations*. Distinct words that correspond to the same symbol are traditionally called *synonyms* if they are in the same language, and *tranlations* if from different languages. Words that may correspond to different symbols are called *ambiguous* (having multiple meanings).

Modelling multiple jurisdictions In the legal context, the distinction of symbols (the meanings) and verbalizations (the words that refer to symbols) allows to model complex situations. Consider the word "murder" (in English) and "Mord" (in German) that are colloquially considered translations. But they are also from different juristictions – common Law vs. German law – so the concepts they refer to are actually slightly different:

Murder in the *common law* is defined as the most serious form of homicide, in which one person unlawfully kills another with the intention to unlawfully cause either death or serious injury [Cr24; Le26b]. In contrast, murder in the *German criminal law* additionally requires at least one of three different variants in reprehensibility-criteria: reprehensible motives, reprehensible manner of commission or reprehensible purpose (see StGB § 211 (2)). Here symbols can come to the rescue: We assume two distinct symbols, e. g., `common-law?murder` vs. `german-law?murder` that are usually referred to by "murder" and "Mord", but we could also have somewhat more explicit verbalizations "Mord im common-law Rechtskreis" and "murder in German law" to refer to them in the respective other language.

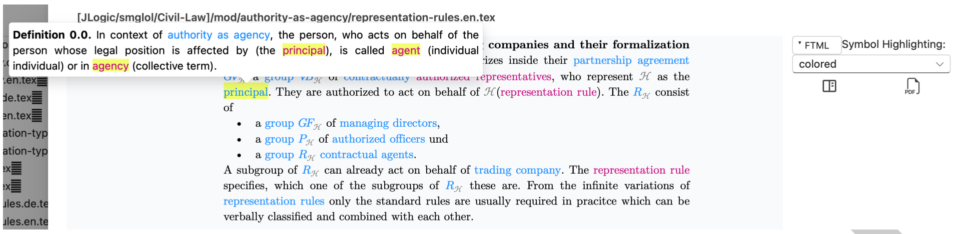

Fig. 1: Representation Rules in FLAMS.

To add insult to injury, the concept of “murder” in US law is subtly different from British law; both juristictions use English, so the word “murder” would technically have to be considered ambiguous. The US law as a whole does not have a singular concept of what a murder is. Many US states have their own definition of murder. *Pennsylvanian criminal law* for instance further differenciates between degrees of murder, e. g., a murder of the first degree, in which the criminal homicide must be commited by an intentional killing (18 Pa.C.S.A. § 2502). We see that the classifications of xxx-nyms become less useful, but symbols can clarify easily: we just introduce a third symbol `US?Pennsylvanian-law?murder` (and maybe a fourth for Canadian murder: `canadian-law?murder`).

Disambiguation by correct annotation Importantly: all we have to do to disambiguate is to annotate the verbalizations with the respective smbols; this can act as a simple (if somewhat tedious; flexiformal) mechanism for fixing the meaning of legal language.

4.3 The Semantic Multilingual Glossary of the Law

Our approach has already enabled legal people in crafting the very first Flexiformal Legal Domain Model by using simple LaTeX-editors, markdown-editors and even Microsoft-Office-Suite with extensions, that links into our Flexiformal Ontology, as mentioned in Sect. 2. For the Semantic Multilingual Glossary of the Law (`SMGloL`) please see `JLogic/SMGloL` in [Ba26b; KW26b]. Even though it is still quite experimental, the `SMGloL` already declares almost 900 symbols in 700 modules, distributed across almost 60 fields of law. Currently, the `SMGloL` mainly focuses on German civil law and thus flexiformalizes in German.

As the `SMGloL` is written in sTeX and thus generates `FTML` which can be served by FLAMS, we can directly see the content at https://mathhub.info/dashboard under `JLogic/SMGloL/Civil-Law/authority-as-agency`; in particular, the contents are served as active documents; see, e. g., Fig. 1 for the concept “Representation Rules” – the rules for deputies’ authority to act on behalf of the company they are in charge of [Ba26a]:

The main body shows the flexiformal definitions (the words highlighted in magenta are the definienda – the technical terms defined here –, and the ones in cyan term references

– technical terms defined elsewhere in the SMGloL and referenced here), and the sidebar gives access to the PDF, the formal annotations (using the the ◫ icon; see Fig. 1), as well as various added-value services (here the color theme; which allows to make the magenta/cyan highlighting invisible or more subtle). This is exactly the information the added-value services can feed on: We see the "example-on-hover feature" in Fig. 2 and the "definition-on-hover" in Fig. 1, here we hovered on the word "Prokurist" and – the hovers are also active documents – then on the word "Prokura".

Note that the SMGloL is intended as a domain model, therefore the definitions are (hand-) crafted to be concise, highly compressed and take advantage of conceptual modularity of the domain, and not to be maximally helpful for the reader – that can be done with in narrative legal documents that reference the SMGloM for grounding. In this way the SMGloL – and the SMGloM it is inspired by – differs from most normative legal documents as well as non-normative legal sources like the Wikipedia or LLM-generated content. A domain model by its very nature is a human-readable flexiformalization of the underlying knowledge: the annotated ontological components (see Fig. 2) drives added value services, and the narrative text allows humans to verify that this is an adequate rendering of the knowledge.

Design choice and explicit implementation Right now in legal, we have decided in a flat bottom-up approach to first distinguish between the different kinds of legal top-domains:

1. *legal general terms*, e. g., different kind of entities like natural and legal persons and what they are;
2. *legal primitives*, e. g., rights, permissions, obligations, and prohibitions;
3. *legal fields*, e. g., civil law, corporate law and even legal philosophy – without embedding specialized fields into general legal fields as this could hurt re-usability.

Then we can start with the definitions, concepts, and structures of single legal terms (the symbols). For each symbol – in its home module – we create a single sTeX-file. Later on we can of course add examples for the symbols (in separate sTeX files, so that they can be separately processed in active documents). The connections between the symbols (the nodes) are established with the additions of (semi-automatic) referential annotation of the definitory text [KS26; Sc26] and the `\usemodule` or `\importmodule` macros from [KM22], which point to other symbols and legal sub-domains. All of this is done in a simple TeX-editor of choice with very few deep technical know-how required. The result is a SPARQL-able (!) holistic network of legal knowledge, i. e., holism [Gi00].

If we want to make things even more formal, we can start implementing functions from the meta-meta-tool of mathematics MMT ([Ra16]) and add types, e. g., permissions, prohibition, and roles. We can even make formally verifiable assumptions/theorems. The latter things still have not been added yet as they require further conceptualization and changes to FLAMS. They are considered future work. This kind of text-editorial formalization process is – at least in the legal domain – yet unique and even for legal people easy to handle. A

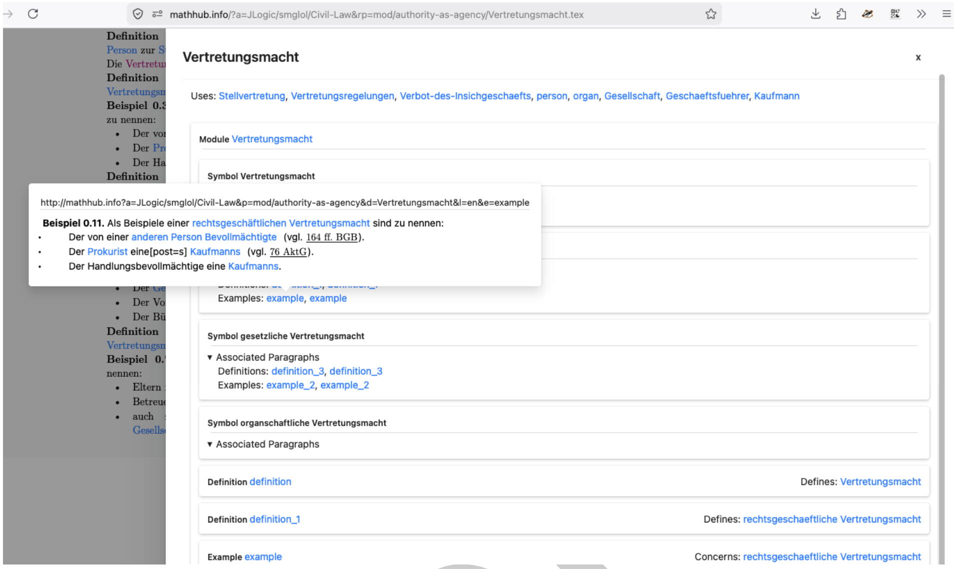

Fig. 2: The Semantic Annotations in Fig. 1

complete and exhaustive implementation of FL∀M∫ would (and is certainly going to) be a huge breakthrough for the legal domain.

5 Conclusion and Future Work

On our current journey inside the DIREGA [DI26] and the FAUStairs [FA26a] research project we have already proven how our Flexiformal Legal Domain Model – the FlexiLDM – has aided the computer scientists in representing the rules for commercial registry applications in a symbolic fashion, e. g., application of new deputies in charge and the corresponding representation rules [DI26] inside `sCASP`, a `Prolog` library for humans natural readable outputs. For the initial paper towards `sCASP` see [Ar18]. For a prettified explanatory document on a `sCASP`-verified application of a German Ltd.'s office address change see `smglom/legal/registry-law/commercial-registry/applications/GmbH` on MathHub.

Scalability One major concern in the domain modelling process is of course scalability: We at KWARC already have tools that (semi-) automate the annotation process itself [Sc26; FA26b; KS26]. On top of that, it is still to be researched, how well knowledge engineers can be supported in their rigorous- to flexiformlization process – which is still very novel on its own – through the use of LLM-agents. E.g., by generating bare-bone modules and pre-filling in necessary concepts while restricting this process to certain input data like educational material or glossaries [Le26b].

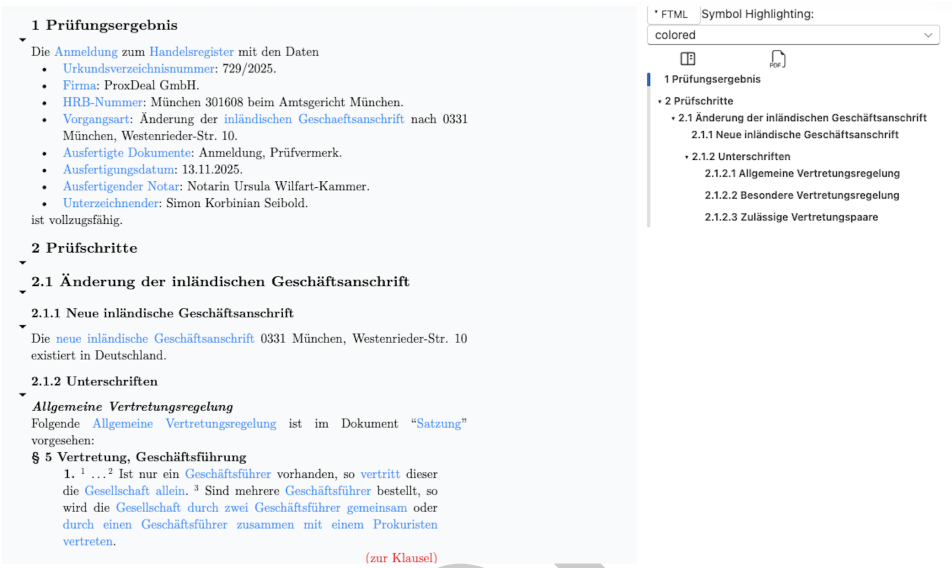

Fig. 3: Explanation of a registry application's verification in FLAMS.

Public services Regardless of scalability, this paper shall and does already serve to finally facilitate and provide public services similar to open-sourced public E-Learning support systems such as ALeA [Be23; CKM23; KW26a] with corresponding case and law representation, incl. interpretational views and theories similar to [KRS25; CFK14] and supporting the citizens in exercising their rights including easily understandable "guided tours" through the legal materia, cf. a semantic representation of mathematics behind [KW26a; BKM24]; for suggested semantics services see [SP08; So09]. A corresponding first-party funding application for the hereby suggested – or rather hinted – E Learning System "ALeA Jus" by the authors' intuition has already been filed together with Prof. Dr. Paulina Jo Pesch, Assistant Professor of Civil Law, Law of Digitization, Data Protection Law and Law of Artificial Intelligence.[5]

[5] FAU Erlangen-Nürnberg, Law School, Schillerstr. 1, 91058 Erlangen, paulina.pesch@fau.de, 0000-0002-5508-2918.

References

[AB23] Atkinson, K.; Bench-Capon, T.: ANGELIC II: An Improved Methodology for Representing Legal Domain Knowledge. In: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law. ICAIL '23, Association for Computing Machinery, Braga, Portugal, pp. 12–21, 2023, https://doi.org/10.1145/3594536.3595137.

[Ad25] Adrian, A. et al.: Maschinelles Schliessen mit s(CASP) - Anmeldung eines neuen Geschäftsführers einer GmbH zum Handelsregister. Jusletter IT, pp. 433–442, 2025.

[Ar18] Arias, J. et al.: Constraint Answer Set Programming without Grounding, 2018, arXiv: 1804.11162 [cs.PL], https://arxiv.org/abs/1804.11162.

[Ba26a] Basaran, A.: Representation Rules, online, 2026, https://mathhub.info/?a=JLogic/smglol/Civil-Law&rp=mod/authority-as-agency/representation-rules.en.tex.

[Ba26b] Basaran, A.: Semantic Multiglingual Glossary of Law, online, 2026, https://mathhub.info/?a=JLogic/smglol.

[Be23] Berges, M. et al.: Learning Support Systems Based on Mathematical Knowledge Management. In (Dubois, C.; Kerber, M., eds.): Intelligent Computer Mathematics. Springer Nature Switzerland, Cham, pp. 84–97, 2023.

[BKM24] Betzendahl, J.; Kohlhase, M.; Müller, D.: Guided Tours in ALeA. In (Nowaczyk, S. et al., eds.): Artificial Intelligence. ECAI 2023 International Workshops. Springer Nature Switzerland, Cham, pp. 397–408, 2024.

[CB91] Coenen, F.; Bench-Capon, T.: Exploiting isomorphism: development of a KBS to support British Coal insurance claims. In (susskind richard, r., ed.): Proceedings of the 3rd International Conference on Artificial Intelligence and Law. Pp. 62–68, 1991.

[CFK14] Carette, J.; Farmer, W.; Kohlhase, M.: Realms: A Structure for Consolidating Knowledge about Mathematical Theories. Lecture Notes in Computer Science, 2014.

[CG78] Cappelletti, M.; Garth, B.: Access to Justice Vol. I: A world survey (Book I and II). The Florence Access-to-Justice Project. Giuffre. Sijthoff and Noordhoff, 1978.

[CGJ77] Cappelletti, M.; Gordley, J.; Johnson, E.: Toward Equal Justice: A Comparative Study of Legal Aid in Modern Societies (Texts and Materials). The Cambridge Law Journal 36 (1), ed. by Munday, R., pp. 174–175, 1977.

[CKM23] Chugh, A.; Kohlhase, M.; Müller, D.: Presentation of Active Documents in ALeA. 2023, https://cicm-conference.org/2023/mathui/mathuiPubs/CICM_2023_paper_1615.pdf.

[Cr24] Crown Prosecution Service: Homicide: Murder, manslaughter, infanticide and causing or allowing the death or serious injury of a child or vulnerable adult, online, 2024, https://www.cps.gov.uk/prosecution-guidance/homicide-murder-manslaughter-infanticide-and-causing-or-allowing-death-or#a03.

[DI26] DIREGA research group: Digitaler Registerassistent, 2026, https://www.direga.fau.de/en/.

[Ea96] Easterbrook, F. H.: Cyberspace and the Law of the Horse. University of Chicago Legal Forum, Precursor to Lessig's work. Argues cyberlaw may be unnecessary, pp. 207–216, 1996.

[Eu26] European Data Protection Board (EDPB): Guidelines, Recommendations, Best Practices, 2026, https://www.edpb.europa.eu/our-work-tools/general-guidance/guidelines-recommendations-best-practices_en.

[FA26a] FAUStairs research group: Formative Assessment for Universities: Strategic Application of Innovative Methods to Raise Study Success Rates, 2026, https://www.faustairs.fau.de/en/.

[FA26b] FAUStairs research group: GloX. The FAUStairs Glossary Extractors and Curator, 2026, https://github.com/KWARC/GloX.

[FH18] Filippi, P. D.; Hassan, S.: Blockchain Technology as a Regulatory Technology: From Code is Law to Law is Code. CoRR abs/1801.02507, 2018, arXiv: 1801.02507, http://arxiv.org/abs/1801.02507.

[FI26] FITKO (Föderale IT-Kooperation): FIM. Föderales Informationsmanagement. Der Standard für Verwaltungsleistungen. online, Translation: Federal Information Management. The standard for public administration services., 2026, https://www.fitko.de.

[FLL22] Flatt, A.; Langner, A.; Leps, O.: Model-Driven Development of Akoma Ntoso Application Profiles. A Conceptual Framework for Model-Based Generation of XML Subschemas. Springer Cham, 2022.

[Gi00] Gibson, R. F.: Quine, Wittgenstein, and Holism. In (Orenstein, A.; Kotatko, P., eds.): Knowledge, Language and Logic: Questions for Quine. Springer Netherlands, Dordrecht, pp. 81–93, 2000, https://doi.org/10.1007/978-94-011-3933-5_7.

[Go20] Governatori, G. et al.: 'Rules as Code' will let computers apply laws and regulations. But over-rigid interpretations would undermine our freedoms, online, 2020, https://theconversation.com/rules-as-code-will-let-computers-apply-laws-and-regulations-but-over-rigid-interpretations-would-undermine-our-freedoms-149992.

[Jü21] Jünemann, M.: Can Code Be Law? 2021, https://www.twobirds.com/en/insights/2021/germany/can-code-be-law.

[KJS24] Kim, M.; Jackson, D.; Sievert, J.: Legal Design, Dignifiying People in Legal Systems. Cambridge University Press, 2024.

[KM22] Kohlhase, M.; Müller, D.: System Description STEX3–A LATEX-Based Ecosystem for Semantic/Active Mathematical Documents. In: International Conference on Intelligent Computer Mathematics. Springer, pp. 184–188, 2022.

[Ko08] Kohlhase, A.: Semantic Interaction Design. Composing Knowledge with CPoint, 2008, https://kwarc.info/people/akohlhase/pubs/AKo_Promo.pdf.

[Ko13] Kohlhase, M.: The Flexiformalist Manifesto. In. Pp. 30–36, 2013, https://kwarc.info/kohlhase/papers/synasc13.pdf.

[Ko22] Kohlhase, M.: Wann ist ein juristischer Text strukturiert? Die Sicht der Informatik und der Künstlichen Intelligenz. In. Pp. 155–170, 2022, https://kwarc.info/kohlhase/papers/wids22.pdf.

[Ko26a] Kohlhase, M.: Flexiformal Ontology, online, 2026, https://mathhub.info.

[Ko26b] Kohlhase, M.: Semantic Multiglingual Glossary of Math, online, 2026, https://mathhub.info/?a=smglom.

[KRS25] Kohlhase, M.; Rabe, F.; Schütz, M.: Lightweight Realms. In (de Paiva, V.; Koepke, P., eds.): Intelligent Computer Mathematics. Springer Science and Business Media Deutschland GmbH, pp. 222–239, 2025.

[KS26] Kohlhase, M.; Schaefer, J. F.: Semantic Authoring in a Flexiformal Context — Bulk Annotation of Rigorous Documents. In (de Paiva, V.; Koepke, P., eds.): Intelligent Computer Mathematics. Springer Nature Switzerland, Cham, pp. 208–221, 2026.

[KW26a] KWARC research group: ALeA. Adaptive Learning Assistant. E-Learning Platform, 2026, https://alea.education.

[KW26b] KWARC research group: MathHub. A portal for active mathematical documents and an archive for flexiformal mathematics, online, Also see mathhub.info/., 2026, kwarc.info/systems/mathhub/.

[Le00] Lessig, L.: Code is Law. Harvard Magazine 102, Popularization and clarification of Code is Law thesis, pp. 16–18, 2000, https://www.harvardmagazine.com/2000/01/code-is-law-html.

[Le26a] Legal Information Institute at Cornell Law School: Wex, LII's community-built, freely available legal dictionary and legal encyclopedia, 2026, https://www.law.cornell.edu/wex/.

[Le26b] Legal Information Institute of Cornell Law School: Wex Glossary, online, 2026, https://www.law.cornell.edu/wex/.

[Le99] Lessig, L.: Code and Other Laws of Cyberspace. FOUNDATIONAL: Introduces "Code is Law" concept. Argues code regulates behavior like law, Basic Books, 1999.

[Li23] Libal, T.: Legal linguistic templates and the tension between legal knowledge representation and reasoning. English, Frontiers in Artificial Intelligence 6, 2023, https://www.frontiersin.org/articles/10.3389/frai.2023.1136263/full.

[Li24] Lindqvist, A.: The thing with Rules. 2024, https://interoperable-europe.ec.europa.eu/collection/eugovtech/news/rules-code-rac.

[Lo23] Lohr, D. et al.: The Potential of Answer Classes in Large-scale Written Computer-Science Exams. In: Proceedings of the 10th Symposium on Computer Science in Higher Education HDI23. Accepted, Aachen, Germany, 2023, https://kwarc.info/kohlhase/submit/HDI23-answerclasses.pdf.

[MCP21] Merigoux, D.; Chataing, N.; Protzenko, J.: Catala: a programming language for the law. Proceedings of the ACM on Programming Languages 5 (ICFP), pp. 1–29, 2021, http://dx.doi.org/10.1145/3473582.

[Mo19] Mohun, J.: Rules as Code: Seeking contributions, online, 2019, https://oecd-opsi.org/blog/new-rules-contributions/.

[Mo20] Morris, J.: Blawx: Rules as Code Demonstration, online, 2020, https://law.mit.edu/pub/blawxrulesascodedemonstration/release/1.

[Mo25a] Morris, J.: Blawx WebApp, online, In development since 2018, 2025, https://app.blawx.dev/.

[Mo25b] Morris, J.: L4 - Rules as Code, online, 2025, https://l4.legalese.com.

[MR20a] Mohun, J.; Roberts, A.: Cracking the code: Rulemaking for humans and machines, OECD Working Papers on Public Governance 42, OECD Publishing, 2020, https://ideas.repec.org/p/oec/govaaa/42-en.html.

[MR20b] Mohun, J.; Roberts, A.: Report Launch – OPSI Innovation Primer on Rules as Code, online, 2020, https://oecd-opsi.org/blog/report-launch-opsi-innovation-primer-on-rules-as-code/.

[Mü26] Müller, D.: The Flexiformalist Manifesto Made Manifest. 2026, https://mathhub.info/?a=Papers/26-BAI-FLAMS&rp=paper.tex.

[Na08] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf, Original Bitcoin whitepaper. Foundation for Code is Law in cryptographic systems, 2008.

[Ne18] of New Zealand, G.: Better Rules for Government Discovery Report, online, 2018, https://www.digital.govt.nz/dmsdocument/95-better-rules-for-government-discovery-report/html.

[Os21] Oster, J.: Code is code and law is law—the law of digitalization and the digitalization of law. International Journal of Law and Information Technology 29 (2), pp. 101–117, 2021, eprint: https://academic.oup.com/ijlit/article-pdf/29/2/101/40275146/eaab004.pdf, https://doi.org/10.1093/ijlit/eaab004.

[Pa11] Palmirani, M. et al.: LegalRuleML: XML-Based Rules and Norms. In (Olken, F.; Palmirani, M.; Sottara, D., eds.): Rule-Based Modeling and Computing on the Semantic Web. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 298–312, 2011.

[Qu22] Quinn, J.: 'Code Is Law' During The Age Of Blockchain. Forbes Business Council, 2022, https://www.forbes.com/councils/forbesbusinesscouncil/2022/05/17/code-is-law-during-the-age-of-blockchain/.

[Ra16] Rabe, F.: MMT: A Foundation-Independent Approach to Formal Knowledge, online, Also see https://github.com/open-mpi/mtt/., 2016, https://uniformal.github.io/doc/philosophy/articles/mmt.pdf.

[Ra25] Raupach, M. et al.: Towards a Semantic Format for FIM: Supporting German Public Services Using the GerPS-FIM- Microverse Ontology Pipeline. In (Asghari, H.; Züger, T., eds.): Proceedings of the 2nd Workshop on Public Interest AI (PI-AI 2024) co-located with the German Conference on AI (KI 2024). September 23, 2024, Julius-Maximilians Universität Würzburg, Germany. 2025, https://ceur-ws.org/Vol-3958/piai24-paper3.pdf.

[Ro20] Robaldo, L. et al.: Formalizing GDPR Provisions in Reified I/O Logic: The DAPRECO Knowledge Base. Journal of Logic, Language and Information 29, 2020.

[Ru25] Rulemapping Group GmbH: Rulemapping. Rule AI, online, 2025, https://rulemapping.com/.

[Sc26] Schäfer, F.: Snify, 2026, https://github.com/slatex/stextools.

[So09] Sonntag, M.: Generating Cases for E-Learning. In (Governatori, G., ed.): Legal Knowledge and Information Systems. Jurix 2009: The Twenty-Second Annual Conference. IOS Press, 2009.

[SP08] Sonntag, M.; Paramythis, A.: Adaptive Feedback for Legal E-Learning. In: Proceedings of the special track. Computer-based Knowledge and Skill Assessment and Feedback in Learning Settings at the International Conference on Computer-Aided Learning, Villach. Kassel university press, 2008.

[Sz94] Szabo, N.: Smart Contract. 1994, https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

[Va24] Vašíček, O. et al.: Early Validation of High-level System Requirements with Event Calculus and Answer Set Programming, 2024, arXiv: 2408.09909 [cs.LO], https://arxiv.org/abs/2408.09909.

[Wa21] Waddington, M.: Rules as Code. Law in Context. A Socio-legal Journal 37, pp. 179–186, 2021.

[Zh20] Zheng, Z. et al.: An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems 105, pp. 475–491, 2020, http://dx.doi.org/10.1016/j.future.2019.12.019.

[ZRH21] Zinovyev, E.; Reule, R. C.; Härdle, W.: Understanding Smart Contracts: Hype or hope? IRTG 1792 Discussion Paper No. 2021-004, 2021, https://www.econstor.eu/handle/10419/233510.