Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220469

An Argumentation and Ontology Based Legal Support System for AI Vehicle Design

Yiwei LU [a,1,2], Zhe YU [b,3], Yuhui LIN [c], Burkhard SCHAFER [a], Andrew IRELAND [c], Lachlan URQUHART [a]

[a] *School of Law, Old College, University of Edinburgh*
[b] *Institute of Logic and Cognition, Department of Philosophy, Sun Yat-sen University*
[c] *School of Mathematical and Computer Sciences, Heriot-Watt University*
e-mail: Y.Lu-104@sms.ed.ac.uk, zheyusep@foxmail.com, {B.Schafer, lachlan.urquhart}@ed.ac.uk, {y.lin,ceeai}@hw.ac.uk

Abstract. As AI products continue to evolve, increasingly legal problems are emerging for the engineers that design them. Current laws are often ambiguous, inconsistent or undefined when it comes to technologies that make use of AI. Engineers would benefit from decision support tools that provide engineer's with legal advice and guidance on their design decisions. This research aims at exploring a new representation of legal ontology by importing argumentation theory and constructing a trustworthy legal decision system. While the ideas are generally applicable to AI products, our initial focus has been on Autonomous Vehicles (AVs).

Keywords. Legal ontology, Autonomous vehicle, Legal detection, Argumentation theory, Explainable AI

1. Introduction

Concerns about the safety of AVs, and a recognition that widespread lack of trust in them will impede their uptake, have resulted in a plethora of legislative and regulatory activity such as the EU AI Act [1], or, specifically for AVs, the proposal by the Law Societies of England and Wales. Our contention is that legal AI, and more specifically a combination of legal ontologies and argumentation systems, can help alleviate these compliance burdens by providing intelligent design environments that help the engineer to reason through the legal implications of their design choices.

A similar approach has been developed as part of the Smarter Privacy project that aims at assisting developers of smart grids to comply with data protection law [2]. Their approach modelled the subject domain using the Sumo ontology, enriched with concepts from data protection law, and combined it with a rule-based reasoner about the relevant

[1]Work supported by UKRI Research Node on Trustworthy Autonomous Systems Governance and Regulation (EP/V026607/1, 2020-2024).
[2]For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.
[3]Corresponding Author.

legal domain. While sympathetic to this approach, our proposal differs in a crucial jurisprudential assumption that we think implicitly underlies their solution: there the law and its categories are taken as a given, and the reasoner then merely subsumes new facts under the old categories. The result is a "Dworkinian one-right-answer" [3] approach. By contrast, we argue that the legal analysis of new technologies takes place under uncertainty not just of the facts but also the law, whose categories can become unstable in response to external change, contested and open to revision.

Our approach, in a nutshell, is this: the introduction of AVs and other autonomous systems creates fundamental challenges to the legal system that can no longer be resolved by mere analogous application of existing categories to these new objects. Rather, they potentially "break" the underlying ontology and conceptual divisions of the law, creating systematic inconsistencies and gaps, which are then in need of "ontology repair". Because law, like language, is self-reflective, this process of ontology repair in turn uses legal arguments in one and the same decision, the judge may e.g. propose an interpretation that subsumes the facts under an existing legal category, while also making an argument that some higher-order legal principle requires to be amended, deleted or added to the existing categories. This ability of lawyers to reason *about* legal categories in addition to *using* them is particularly visible when more fundamental changes in the external world create gaps, inconsistencies and ambiguities when old categories are applied to new realities. These exercises in ontology repair and ontology evolution inevitably create legal uncertainty. As we will see in the examples below, this can create an unmet need for engineers (or other members of society) to make legally informed decisions under uncertainty. We aim to show how building on existing approaches to legal ontologies and legal reasoning that make ontology repair explicit can help to address this.

A simple example may help to explain this more abstract notion. The United Kingdom Department of Transport 2015 report *The Pathway to Driverless* Cars stated that - testing of automated technologies is legally possible, provided that *the vehicle can be used compatibly with road traffic law*. In other words, the AV must observe the same rules originally addressed to human drivers. How can a developer of an AV make sense of this requirement? A starting point would be to consult the relevant road traffic rules, and treat the AV as the new norm addressee that "inherits" the legal obligations of the human driver. For some of these, this change is unproblematic and merely reinterprets the old category of "driver" as including "autonomous vehicles". For other rules, however, this strategy is less convincing. A candidate could be the rule that "the driver must not be drunk". Here the engineer can continue to treat the AV as "the driver", and as cars are never drunk, the conditional norm: "If drunk, don't drive" is trivially true all the time, and the car is trivially compliant with this provision. Alternatively, "the driver" in this context may refer to some human inside the car who may have been assigned specific legal duties like healing the injury in accidents. This means the concept of "driver" has now been subdivided to "heal" the counter-intuitive outcome. If this interpretation is taken, a number of follow-up questions need to be answered. In one interpretation, this human "non-driver" is responsible for being sober and faces sanctions when drunk but this is not a concern of the car developer. However, another interpretation is also possible: here, the duty to ensure that the vehicle is operated lawfully transfers more fully to the AV, which now has to monitor if there is at least one sober passenger, e.g. [4].

The underlying problem that leads to these three different interpretations is that AVs share some properties with the category "driver" and some properties with the disjoint category “car”, creating systematic ambiguities when interpreting laws whose semantics reflect the old ontology. Even more fundamentally, the reason AI regulation is difficult is that they seem to violate some of the most basic ontological distinctions that structure the law, in particular the distinction between persons and objects. This was a central point made by the joint report of the Law Commissions of England and Scotland. They note that “Existing law reflects a division between rules governing vehicle design on the one hand and the behaviour of drivers on the other.” What the Commissions ask for in response is a new conceptual scheme that bridges these two regulatory spheres: the automated driving system is at the same time equipment fitted in a vehicle (and object) but it also determines the behaviour of the vehicle (an agent). We can repurpose research in ontology repair to model not only the reasoning of the Commission, but also how that document can in turn be made into a legal argument that informs design decisions. Ideally, at every decision that they have to document, the engineers need a system that

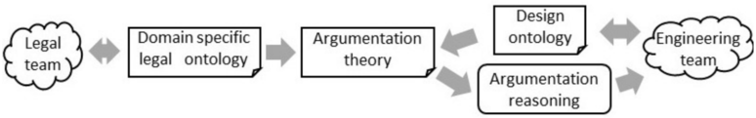

Figure 1. Overview of the process of legal support system

(1) Gives feedback about whether a design draft is in compliance with current or possible laws, depending on which of several competing interpretations is chosen, (2) Answers what happens for the legal analysis if a single functionality is added, deleted or modified in current design draft; (3) Supports reasoning based on how conflicting preferences and values have been resolved; (4) Gives an understandable explanation of the legal results for auditing purposes. Here, we present a legal support system for autonomous vehicles (*LeSAC*), as shown in Figure 1. It is built on top of legal ontology and a legal reasoning based legal argumentation framework, i.e. *L-ASPIC* [5], adapted from [6].

Legal ontologies have proved to be very strong tool for law as legal expert systems, legal databases and documents management: There are different functions that have been proposed, e.g. the Legal Knowledge Interchange Format (LKIF) Core ontology builds on the Web Ontology Language (OWL) and rules [7]; LRI-Core aimed at the legal domain grounded in common-sense [8] and UOL [9]. And there are legal ontology models constructed for specific legal domains like ALLOT[10] etc.

However, a legal ontology alone is insufficient for legal reasoning. The main description language of the current legal ontology is the Web Ontology Language (OWL) or OWL2, whose semantics are based on DL [11] and they cannot support inconsistent reasoning as a subset of first-order logic, which is quite important in legal reasoning. This is also a reason that most of the existing legal ontology focuses more on capturing abstract legal concepts, playing the role of document management or legal dictionary. To address this problem, there are works focused on detecting and repairing inconsistent parts [12] or extending classical logic by adding true values [13]. However, these works weaken the reasoning strength of DL [13] and require guidance outside the ontology [12]. Also, they lack explainability which is a desirable feature when inconsistency happens.

There are other works taking a formal argumentation approach. Formal argumentation has been noticed as an approach to dealing with reasoning under inconsistent and uncertain contexts [14,15]. Formal argumentation has the merits of computational efficiency and explainability [16,17]. For handling reasoning with inconsistent ontologies, several previous studies have considered applying structured argumentation systems in this field, e.g. [15,18]. These works support inconsistent reasoning but they cannot describe more complicated interactions or agents' different attitudes. And they also do not discuss the explanation of reasoning results or design the legal semantics and functions in their reasoning processes.

The rest of this paper is organized as follows. §2 firstly introduce a running example, as well as an introduction of DL and argumentation theory. Then it briefly describes how the *LeSAC* is deigned and shows some of its important functions. §3 concludes this paper.

2. *LeSAC* and a case study

To help clarification, we start with a very possible scenario in the future, of which the coding could be found in [19].

Example V1. *Currently, the law stipulates a number of behaviours that a human driver has to observe after an accident has happened. This includes a duty to stay at the scene of an accident and to provide first aid if necessary and feasible. Let's assume there is one passenger in the car, but he is (illegally) too drunk to do anything. In such a "contrary to duty" scenario, how should the AV car react now when somebody is hit?*

To handle this possible situation, we refer to current and relevant legal rules. We extract and select some most relevant information from traffic law and criminal law:

(1) It is illegal to drive a motor vehicle while intoxicated. People who drive while intoxicated shall lose their driving license and may be prosecuted in criminal law.

(2) A person who commits a hit-and-run accident will be criminal responsibility, especially when the escape causes the death or the driver is intoxicated.

(3) When an accident happens, the driver should take the responsibility to transfer the injured party to a safe place and provide aid if the situation is urgent.

(4) It's illegal to let a drunk passenger leave the car alone during the trip.

DL is the basic semantic of OWL or OWL2, which are the main logic languages of current legal ontologies. The basic notions of DL systems are *concepts* and *roles*. A DL system contains two disjoint parts: the TBox and the ABox. TBox introduces the terminology, while ABox contains facts about individuals in the application domain. There are many DLs and this paper is based upon the *ALC* expression [20,11].

2.1. Legal support system for autonomous vehicles

In [5] we constructed a structured argumentation framework *L-ASPIC* for reasoning based on an inconsistent legal ontology. Then given a legal ontology, particularly for AVs design, we can construct a *LeSAC* system based on *L-ASPIC* . Based on *LeSAC* , an argumentation framework for example V1 will be like:

Example V2.

$$\mathcal{N} = \left\{ \begin{array}{l} r_1 : Driver(x) \Rightarrow Sober(x); \\ r_2 : Intoxicated(x) \Rightarrow \neg LeaveCar(x); \\ r_3 : Driver(x), Intoxicated(x) \Rightarrow BeRevokedDrivingLicense(x); \\ r_4 : Driver(x), Intoxicated(x) \Rightarrow TakeCriminalResposibility(x); \\ r_5 : hitAndRun(x,y) \Rightarrow TakeCriminalResposibility(x); \\ r_6 : hitAndRun(x,y), causeDeath(x,y) \Rightarrow AggravatedPunishment(x); \\ r_7 : hitAndRun(x,y), Driver(x), Intoxicated(x) \Rightarrow AggravatedPunishment(x); \\ r_8 : CauseAccident(x), Injury(y) \Rightarrow transferToSafePlace(x,y); \\ r_9 : CauseAccident(x), Injury(y), NeedEmergencyAid(y) \Rightarrow doNecessaryAid(x,y) \end{array} \right\}$$

$$\mathscr{R}_s{}^{4} = \left\{ \begin{array}{l} r_{10} : Sober(x) \rightarrow \neg Intoxicated(x); \\ r'_{10} : Intoxicated(x) \rightarrow \neg Sober(x); \\ r_{11} : transferToSafePlace(x,y) \rightarrow LeaveCar(x); \\ r'_{11} : \neg LeaveCar(x) \rightarrow \neg transferToSafePlace(x,y); \\ r_{12} : doNecessaryAid(x,y) \rightarrow LeaveCar(x); \\ r'_{12} : \neg LeaveCar(x) \rightarrow \neg doNecessaryAid(x,y) \end{array} \right\} \quad \mathscr{K}^A = \left\{ \begin{array}{l} Driver(PS1); Intoxicated(PS1); \\ hitAndRun(PS1, Injury1); \\ Injury(Injury1); \\ causeDeath(PS1, Injury1); \\ CauseAccident(PS1); \\ NeedEmergencyAid(Injury1) \end{array} \right\}$$

$$\mathscr{P} = \left\{ \begin{array}{l} p_1 : \textit{Human lives should be protected as a priority}; \\ p_2 : \textit{AI products should avoid extra risk about safety for their users}; \\ p_3 : \textit{People should avoid putting others into dangerous by his own behaviours,} \\ \qquad \textit{and should bear corresponding responsibility.} \end{array} \right\}$$

$prin(r_1) = p_3$; $prin(r_2) = p_2$; $prin(r_3) = p_3$; $prin(r_4) = p_3$; $prin(r_5) = p_3$;
$prin(r_6) = p_3$; $prin(r_7) = p_3$; $prin(r_8) = p_1$; $prin(r_9) = p_1$

We now present *LeSAC* 's reasoning functions combing the case study.

Legal compliance detection When engineers complete a whole design draft, they could use the consistency checking function to check whether this design is fully compliant with given laws and where conflicts are. If a design is consistent after reasoning, it means it is fully compliant with given laws. Otherwise, it is not. And by tracing where arguments conflict, we could know which part of the design needs modification. Based on the *LeSAC* in Example 2, we can at least construct the following two arguments.

Example (Example V2 cont.)**.**
$\alpha = (CauseAccident(PS1), Injury(Injury1) \Rightarrow transferToSafePlace(PS1, Injury1)) \rightarrow LeaveCar(PS1)$ *and*
$\beta = Intoxicated(PS1) \Rightarrow \neg LeaveCar(PS1)$.

α and β attack each other, therefore the design is not complaint with given laws.

Feedback for single change If the AV engineers want to keep the main design of an AV and only do some minimal changes, *LeSAC* can provide possible further legal consequences with these new details by instance checking. According to *LeSAC* , assertions are the conclusions of arguments. So based on the extension of arguments, we can decide whether an assertion is accepted.

To determine whether a certain modification is consistent with the current design and given laws, we translate this problem into whether a legal assertion about this AV can be accepted as a conclusion of an accepted/justified argument. Considering arguments α and β in Example 2.1, assuming that based on the priority ordering on legal principles ($\mathscr{P}$), α is preferred. Then α can defeat β, but not vice versa. Based on the *LeSAC* in Example 2, no other arguments are conflicting with α. As a consequence, α is sceptically justified as well as the assertion "$LeaveCar(PS1)$" .

Giving legal explanations Considering the needs of AV engineers, the explanation of reasoning results from *LeSAC* should show how a legal conclusion is obtained and which content in this situation makes it accepted or not. For any agent y, we can provide a formal explanation of why a legal conclusion X is accepted for certain design requirement consisting of two parts. One explains how X is reached by presenting all the premises and legal rules contained in $\mathscr{K}^A$ and $\mathscr{N} \cup \mathscr{R}_s$. The other explains why this legal conclusion is accepted by presenting all the legal information and principles applied to construct the arguments that defend α. Considering our running example, for the acceptance of the assertion "$LeaveCar(PS1)$", the explanation is:

[4]Rules r'_{10}, r'_{11} and r'_{12} are the transposed rules of rule r_{10}, r_{11} and r_{12}, respectively.

$$(\{Injury(Injury1), CauseAccident(PS1), NeedEmergencyAid(Injury1)\} \cup \{r_8, r_9\}) \cup \{p_2 < p_1\}$$

and for the acceptance of the assertion "$\neg LeaveCar(PS1)$", it is:

$$(\{Intoxicated(PS1)\} \cup \{r_2\}) \cup (\{Intoxicated(PS1)\} \cup \{r'_{10}\}) \cup \{p_1 < p_2\}$$

3. Conclusion and future work

This paper constructed a legal support system to help engineers of AVs improve legal compliance of the design by importing argumentation theory into legal ontology. In future, we will explore legal representation for importing machine learning, e.g. representation learning. We also plan to integrate it into a conventional engineering workflow.

References

[1] Act AI. Proposal for a regulation of the European Parliament and the Council laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. EUR-Lex-52021PC0206. 2021.
[2] Oberle D. Ontologies and reasoning in enterprise service ecosystems. Informatik Spektrum. 2014;37(4):318-28.
[3] Rosenfeld M. Dworkin and the One Law Principle: A Pluralist Critique. Revue Internationale de Philosophie. 2005;59(233 (3)):363-92.
[4] Zaouk AK, Wills M, Traube E, Strassburger R. Driver alcohol detection system for safety (DADSS)A Status update. In: 24th Enhanced Safety of Vehicles Conference. Gothenburg, Sweden; 2015. .
[5] Yu Z, Lu Y. An Argumentation-Based Legal Reasoning Approach for DL-Ontology. arXiv preprint arXiv:220903070. 2022. Available from: https://arxiv.org/abs/2209.03070.
[6] Prakken H. An abstract framework for argumentation with structured arguments. Argument & Computation. 2010;1(2):93-124.
[7] Hoekstra R, Breuker J, Di Bello M, Boer A, et al. The LKIF Core Ontology of Basic Legal Concepts. LOAIT. 2007;321:43-63.
[8] Breuker J, Valente A, Winkels R. Use and reuse of legal ontologies in knowledge engineering and information management. In: Law and the Semantic Web. Springer; 2005. p. 36-64.
[9] Griffo C, Almeida JPA, Guizzardi G. Towards a legal core ontology based on Alexys theory of fundamental rights. In: Multilingual Workshop on Artificial Intelligence and Law, ICAIL; 2015. .
[10] Barabucci G, Di Iorio A, Poggi F. Bridging legal documents, external entities and heterogeneous KBs: from meta-model to implementation. Semantic Web Journal. 2012.
[11] Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF, editors. The description logic handbook: Theory, implementation, and applications. Cambridge University Press; 2003.
[12] Schlobach S, Cornet R, et al. Non-standard reasoning services for the debugging of description logic terminologies. In: Ijcai. vol. 3; 2003. p. 355-62.
[13] Zhang X, Lin Z, Wang K. Towards a paradoxical description logic for the semantic web. In: International Symposium on Foundations of Information and Knowledge Systems. Springer; 2010. p. 306-25.
[14] Dung PM, Kowalski RA, Toni F. Assumption-Based Argumentation. In: Simari G, Rahwan I, editors. Argumentation in Artificial Intelligence. Boston, MA: Springer US; 2009. p. 100-218.
[15] Gómez SA, Chesñevar CI, Simari GR. Reasoning with Inconsistent Ontologies through Argumentation. Applied Artificial Intelligence. 2010;24(1&2):102-48.
[16] Borg A, Bex F. A Basic Framework for Explanations in Argumentation. IEEE Intelligent Systems. 2021;36(2):25-35.
[17] Čyras K, Rago A, Albini E, Baroni P, Toni F. Argumentative XAI: A Survey. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21); 2021. p. 4392-9.
[18] Bouzeghoub A, Jabbour S, Ma Y, Raddaoui B. Handling conflicts in uncertain ontologies using deductive argumentation. In: WI'17;. p. 65-72.
[19] JURIX-22 paper resource webpage;. Accessed: 2022-09-30. https://colab.research.google.com/drive/1k7KLeDsORPvWBLOg7ySU8NsY9rHSVSMp?usp=sharing.
[20] Schmidt-Schauß M, Smolka G. Attributive concept descriptions with complements. Artificial Intelligence. 1991;48(1):1-26.