From Formal Models to Practical Tools: A Case Study

Katie Atkinson
Joe Collenette*
Trevor Bench-Capon*
{katie,j.m.collenette,tbc}@liverpool.ac.uk
Department of Computer Science
University of Liverpool
Liverpool, UK

Kanstantsin Dzehtsiarou
dzeh@liverpool.ac.uk
Department of Law
University of Liverpool
Liverpool, UK

ABSTRACT

One approach to building legal support systems is to run an executable model of the relevant knowledge through an interface designed to collect information from the user and provide explanations. The usability of such systems depends on the terms used in the law being represented: often only users familiar with the practice and application of the law will be able to provide the required information. Earlier work applied this approach to the European Convention on Human Rights (ECHR). Although the performance of the tool built for that domain was good, the questions posed to the user demanded a good deal of knowledge and experience of the ECHR. Here we use the knowledge of an expert with extensive experience of the ECHR to extend the ADF, through intermediate levels, to identify questions that are appropriate to the target user. We have undertaken a pilot evaluation in which a small number of lawyers have used the prototype program and provided very positive feedback, showing that they are receptive to AI solutions that give effective, explainable decision support.

CCS CONCEPTS

• **Applied computing** → **Law**.

KEYWORDS

ADFs, case-based reasoning, explainability, ECHR

ACM Reference Format:
Katie Atkinson, Joe Collenette, Trevor Bench-Capon, and Kanstantsin Dzehtsiarou. 2021. From Formal Models to Practical Tools: A Case Study. In *ICAIL '21: Eighteenth International Conference on Artificial Intelligence and Law, June 21–25, 2021, Sao Paolo, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Since the pioneering formalisation of the British Nationality Act (BNA) [16], one popular approach to building legal support systems has been to formalise the law and then execute the formalisation through a program able to gather facts about particular cases from the user and provide explanations. In [16] the law in question was statute law, but the approach can also be applied to case law; e.g. [1]. Similar structures are found in hybrid systems based on reasoning with cases such as CABARET [17]. IBP [7] and VJAP [12] which also represent the law as high level rules but instead of directly querying the user, the leaf predicates are resolved according to factors found in precedent cases. In all these approaches, we begin with a high level question, such as '*is Peter a British Citizen?*' and then unfold it through a series of intermediate concepts until a base level is reached. At this point, hybrid systems will launch their case based reasoning mechanism, but if we are using a rule based approach which expects the user to answer to these base level questions, the terms must be readily understood by the user. For a successful system, two kinds of knowledge are required: knowledge as to the *questions that should be asked*, and knowledge as to *how they should be answered.* A formalisation of the legislation provides the first but not the second, and so if the law does not use terms familiar to the users, they will be unable to answer the questions. This is also true of hybrid systems: the users will need to ascribe the factors, which is itself a substantial and not entirely straightforward task. Moreover the knowledge of how to answer the questions or ascribe the factors may vary according to the background and skills of the user answering the questions. This suggests that the formalisation of the law will need to be supplemented by a further sets of intermediate concepts, one for each type of user which will unfold into questions appropriate to the users.

In the case of the BNA, the formalisation of the law resulted in questions such as '*where was Peter born*?' and '*who is the father of Peter?*' which were readily answerable by applicants, or by adjudicators on the basis of an application form. Much of the appeal of this system came from the fact that the system that resulted directly from executing the formalisation was immediately and intuitively usable in both these situations. However, those who followed in their footsteps found that this was not always the case. Problems became apparent in a follow up exercise to the BNA [5]. Two kinds of problems arose:

- First, many of the questions were difficult to answer for, or even, unintelligible to, the lay user, such as '*did Peter pay the qualifying level of contributions in the relevant tax year?*'. Such questions might, however, be answerable by an adjudicator, especially if there was access to the contributions record database. Sometimes the difficulty would be reversed: lay users will know their age, but this may require investigation and verification by the adjudicator, who will therefore need to know what kind of evidence is required. So the questions

ICAIL '21, June 21–25, 2021, Sao Paolo, Brazil

https://doi.org/10.1145/nnnnnnn.nnnnnnn

may need further refinement to enable the target users to answer them, and may need different refinements for different target users. For a lay person, the contributions condition will need to be expressed in terms of working in a particular year, while for the adjudicator age will need to refer further to acceptable forms of proof, such a birth certificates.

- Second, the question may have some particular legal interpretation. For example UK Housing Benefit at one time had an addition if the house was "hard to heat" [4]. While a lay user may well have an opinion on this, in fact there was a very technical definition, spelled out in secondary legislation and case law, in terms of factors such as size and type of house, number of rooms and age of occupants. Clearly this information is both needed to enable lay users to answer the question, and to support the decision making of adjudicators, who may be unfamiliar with the relevant case law.

In [9] the authors provided an executable model of Article 6 of the European Convention on Human Rights (ECHR). However the questions that resulted from that formal model are unlikely to be answered with confidence by a potential applicant, and some might even test an experienced lawyer. To move towards appropriate questions requires knowledge of the theory and practice of the ECHR. In this paper we will discuss what is needed for that model to become usable, in particular for those processing initial applications. Section 2 will briefly describe the ECHR. Section 3 will describe the formal model of [9]. Section 4 will describe the additional analysis we have carried out to move to a usable system, and Section 5 the resulting prototype. Section 6 gives an initial evaluation from members of the intended user group who have used the prototype. Finally Section 7 will offer some concluding remarks.

2 DOMAIN OVERVIEW

The European Convention on Human Rights (ECHR) is a regional human rights treaty that is now ratified by 47 European states and covers almost the whole of Europe. Since 1960, when it was established, the European Court of Human Rights (ECtHR) has delivered judgements in thousands of cases and created a significant body of legal precedents.

The ECHR has proved very popular for experimentation with machine learning techniques for legal judgment predication tasks; for example, see [2], [14], [8], [15] and [13]. These studies all report success, with correct predictions being achieved in around 70-85% of cases. JURI Says [15] reports a success rate of 69% over the last year, although it fell to 60.9% for March 2021[1].

3 ADFS FOR REASONING ABOUT ECHR CASES ON ARTICLE 6

The aim of the work described in [9] was to encapsulate Article 6 in an Abstract Dialectical Framework (ADF) [6]. A program based on the developed ADF predicted whether a particular case was admissible, and if so whether there was a violation Article 6 of the ECHR. One of the main strengths of the program was that it was able to justify its reasoning in terms of the ADF nodes and acceptance conditions, similar to the *how?* explanation of a rule based system. The ADF was created using the ANGELIC methodology [1], designed to capture case law in a manner that supports argumentation techniques and is maintainable. The ANGELIC ADF corresponds to the factor hierarchy of traditional case based systems such as CATO [3]. In an ADF the nodes are connected into a graph structure, and there are acceptance conditions that determine the acceptability of each parent in terms of its children. The ADFs produced by ANGELIC are always trees. In ANGELIC the acceptance conditions take the form of a set of prioritised sufficient conditions, together with a default to make the conditions collectively necessary. Maintainability comes from the modular nature of this structure.

Testing on a limited set of cases achieved a 100% success rate. Although the test set was small, the benefits of the approach were clear; the ADF was able to predict Article 6 cases with a high success rate, and fully justify its reasoning for the predictions. The developed ADF in [9] included a part relating to *admissibility*. The questions developed for admissibility were at a very high level. The model described in this paper, also developed using the ANGELIC methodology, focuses entirely on admissibility. This focus allows a much more detailed analysis to reduce the level of expertise needed to answer the questions with confidence. The decomposition of the high level questions obtained from the documentation used in the [9] does, however, require guidance from someone expert in the relevant statute and case law. We have also replaced the command line prototype of [9] with a visual interface that uses standard interactions to make the program more accessible to non-computer specialists.

4 LEGAL FOUNDATIONS OF THE MODEL

The model described here considers whether an application is admissible or not, which is itself a substantial task. All applications submitted to the ECtHR need to be admissible in order to be considered on merits. In other words, the Court needs to establish that the application complies with a set of formal rules before it can examine the substance of this application [10]. Admissibility includes two types of rules: first, the ECtHR needs to establish that the application falls within its jurisdiction to confirm that the Court can deal with this application. Secondly, the ECHR has established a set of formal rules that the application itself needs to comply with, such as that the application was first submitted at the national level and was rejected by the national judicial bodies and that it should be submitted within 6 months after the highest judicial body rejected the same application on the national level. This application should not be abusive, anonymous or trivial. This application also should not be clearly without merits or - in the ECHR terms – manifestly ill-founded. Again, if these conditions are not satisfied, the Court declares an application inadmissible. The Court's decision as to inadmissibility is final and cannot be appealed against.

The importance of admissibility is often underestimated. On average about 90% of all applications submitted to the ECtHR are declared inadmissible. For instance, in 2019, 44,500 applications were submitted to the Court and in the same year 38,480 applications were declared inadmissible. At the same time, in 2019, the Court delivered only 2,187 meritorious judgments [11]. A large number of applications is declared inadmissible every year, so our project has potential importance for both the applicants who might

[1] JURI Says can be found at https://jurisays.com/ (accessed 2021/03/01).

want to avoid inadmissibility and for the Court for which considering of inadmissible applications takes a significant proportion of its time and resources which could be re-allocated to the meritorious cases and so reduce the backlog. In the next section, we describe the implemented tool that we have produced to enable decision support for the important issue of admissibility of cases submitted to the ECtHR. The model used in the tool captures the factors discussed above that need to be examined to determine admissibility and is a result of close consultation with our expert on the ECtHR.

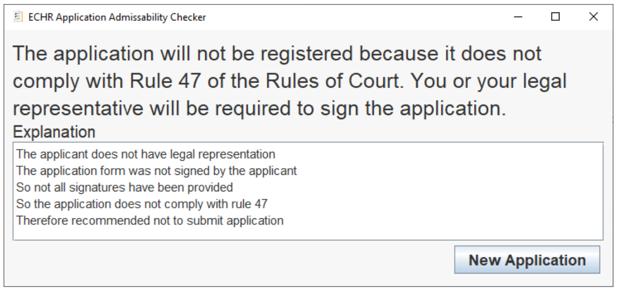
5 PROTOTYPE IMPLEMENTATION

The current prototype consists of two parts; the ADF model, executed by a JAVA program, and the JAVA front-end. The ADF model makes the predictions from a set of answers to the questions gathered from the user through the front-end, which then presents an explanation of the prediction to the user. The aim of the application is to allow people with minimal legal training to quickly identify whether a case that is being submitted to the ECtHR is likely to be admissible. The current ADF is more extensive in the modelling of admissibility when compared to the previous ADF developed in [9]. There are 61 questions which when answered allow the 26 factors to resolve to a prediction. In the previous ADF of [9] there were only 5 factors relating to admissibility and only 7 questions needed to be answered.

When developing the ADF we consulted a legal expert in order to decompose the high level nodes used in [9] and so capture the legal knowledge required to ensure that the list of questions is both as complete as possible, and appropriate to the target users. Table 1 shows the text and the acceptance conditions associated with the nodes. The root of the ADF is "V1" which indicates whether submitting the application to the ECHR is recommended. The program will recommend submission if both issues, nodes I1 and I2, are accepted. That is, the application is admissible (I2) and the application complies with rule 47 of the court (I1). In turn, node I1 is accepted if the base level factors, I1Q1 and I1Q2, are accepted and the abstract factors that represents that all necessary signatures (I1F1) and all documentation (I1F2) have been accepted. Abstract Factor I1F1 has a number of different base level factors, offering different possibilities for I1F1 to be accepted. These represent the different signatures that are needed depending on different situations.

The questions used as part of the ADF are not only based on the requirements of the ECHR but also take into account its case law. The case law makes the questions more nuanced and integrates those aspects of admissibility that do not obviously flow from the text of the Convention. In practice it is almost impossible to submit a successful application without familiarising oneself with the case law, which can be quite broad and diverse, or consulting with a professional lawyer specialising in the law of the ECtHR. This model presents the rules enshrined in the case law of the Court in a simplified form.

In Table 1 I2F2Q4 asks "Is the applicant a potential victim of a violation?" which affects whether the applicant has victim status. It is not obvious to a lay person that potential victims can apply to the Court, in certain limited circumstances. The scope of these circumstances was made clear in the case law: the question has arisen in the context of extradition cases, where the question was whether there was a risk of human rights violations in the receiving state if extradition were permitted e.g. *Soering v. the United Kingdom* (1989). Identifying such questions requires a good knowledge of the legal practice.



The application will not be registered because it does not comply with Rule 47 of the Rules of Court. You or your legal representative will be required to sign the application.

Explanation

The applicant does not have legal representation
The application form was not signed by the applicant
So not all signatures have been provided
So the application does not comply with rule 47
Therefore recommended not to submit application

Figure 1: The results screen of the JAVA prototype. The user is presented with a high level result and the reasoning behind the decision is shown in the text below.

When the JAVA program loads, it parses a text representation of the ADF, similar to Table 1. This supports maintenance by enabling any changes to the ADF to be automatically reflected in the program. The program asks the user a series of yes/no questions and once all the relevant questions have been answered the results are shown to the user. The results screen also displays the reasoning as to why that decision was made. Figure 1 shows the results for an example where the program informs the user that a signature is required for the form and explains that as the applicant does not have legal representation and they have not signed the application form, not all signatures have been provided, which in turn means that the application does not comply with rule 47 of the rules of the court and therefore the program recommends not to submit the application.

To give a better user experience than the previous work [9] only questions that are needed to generate a recommendation are asked: if a node can be resolved, the program moves on to the next. Different paths which lead to different questions; the questions needed if the application is submitted by a company rather than as an individual are notably different.

Using the example in Figure 1 and referring to Table 1, we will show how the explanation is generated in Figure 1. The last question to be answered is I1F1Q2. We start by printing all the base level factors that have the same parent as the last base level factor, as long as the user has answered the corresponding question. The parent of I1F1Q2 is I1F1, and the only other base level factor that has an answer is I1F1Q1. This gives us the first two lines of explanation. Next we print the abstract factors and issues in a hierarchical manner, adding the word "So" before each explanation. Finally we print the root and add the word "Therefore" at the start of the root's explanation.

This prototype has a number of benefits when compared to the system in [9]. Despite the two systems tackling two different aspects of legal reasoning within the ECHR domain, they are comparable as both systems use an underlying ADF that is presented to a user. The major benefit the current approach is the ease of use when compared to the previous work. We have now incorporated both

Table 1: Nodes of the ADF that are referenced, along with their acceptance conditions.

ID	Factor Text	Accepting Logic
V1	Submission Recommendation	ACCEPT IF I1 AND I2 REJECT OTHERWISE
I1	Application compiles with Rule 47	ACCEPT IF I1Q1 AND I1Q2 AND I1F1 AND I1F2 REJECT OTHERWISE
I1Q1	Has the applicant identified the state against which the application is brought to the Court (p2 of the application form)?	ACCEPT IF TRUE REJECT OTHERWISE
I1Q2	Has the applicant ticked an appropriate box on p2 of the application form?	ACCEPT IF TRUE REJECT OTHERWISE
I1F1	All Signatures Provided	ACCEPT IF (I1F1Q1 AND I1F1Q4 AND (I1F1Q2 OR I1F1Q3)) OR (!I1F1Q1 AND I1F1Q3) OR (I1F1Q8 AND ((!I1F1Q1 AND I1F1Q7) OR (I1F1Q1 AND I1F1Q5 AND (I1F1Q6 OR I1F1Q7)))) REJECT OTHERWISE
I1F1Q1	Does the applicant have a legal representative?	ACCEPT IF TRUE REJECT OTHERWISE
I1F1Q2	Was the application form signed (p13) by the applicant's legal representative?	ACCEPT IF TRUE REJECT OTHERWISE
I1F2	All Documentation	ACCEPT IF I1F2Q1 AND I1F2Q2 REJECT OTHERWISE
I2	Application is Admissible	ACCEPT IF I2F1 AND I2F2 AND I2F4 AND I2F5 AND (I2F3F1 AND I2F3F2 AND !I2F3F3 AND I2F3F4 AND I2F3F5) REJECT OTHERWISE
I2F2	Victim Status	ACCEPT IF !I2F2Q3 AND (I2F2Q1 OR I2F2Q2 OR I2F2Q4) REJECT OTHERWISE
I2F2Q4	Is the applicant a potential victim of a violation?	ACCEPT IF TRUE REJECT OTHERWISE

knowledge of the law, and how that law is applied in practice, which allows expression in terms of questions which the users can be expected to answer . For example, in [9] there is a question "Was the person bringing the case the victim?". To answer the question effectively the user needs to understand what constitutes a person, a case, and a victim. The current ADF has decomposed thr base level factor into abstract factors such as I2F1 which describes what constitutes a valid petitioner, and I2F2 which describes victim status (see Table 2 for the questions associated with these abstract factors). The questions associated with these abstract factors require less specialist knowledge. Improvements have been made to the method of interacting with the application, which is more in line with what non-specialist computer users would expect from an application. The previous work used a text-only, command line, interface [9]. The new application uses a visual,mouse-driven, interface.

6 PILOT STUDY

Before embarking upon a full evaluation, we have conducted a pilot study in which the prototype has been tested by a sample of our target audience, which is a small group of lawyers who work within the ECHR. Three independent lawyers (not, of couse, including our domain expert) tested the prototype and completed a questionnaire that covering five different aspects of the prototype.

The following is the list of questions that the users of the prototype were presented with, along with the question category in brackets. Each question had four possible responses, for different levels of agreement:

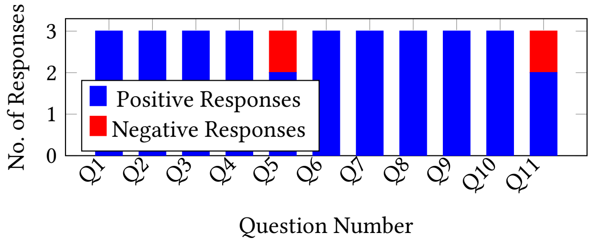


Figure 2: Graph showing number of feedback responses to each question, as positive or negative feedback

(1) (Functionality) Does the program have a reasonable response time?
(2) (Functionality) Did the program run to completion without any interruptions?
(3) (Usability) How easy was the program to use?
(4) (Usability) How intuitive was the program to start using?
(5) (Explainability) How effective was the explanation given for describing the program's decisions?
(6) (Explainability) How easy was the information to parse?

(7) (Usefulness) How useful would you find this program for assisting you in your work?
(8) (Usefulness) Generally how useful would additional technology be for assisting with legal work?
(9) (Questions about Questions) How clear were the questions that you answered within the program? Additionally the users were able to say which questions were unclear.
(10) (Questions about Questions) How much time would you save if you used a fully functional program for your work on deciding on the admissibility of cases?
(11) (Questions about Questions) Does the program reflect how you decide on the admissibility of cases that you process?

In Figure 2 the responses to the questionnaire have been condensed into positive or negative responses, where the top two answers to a question are positive and the bottom two are negative.

Though the results of the questionnaire come from a very small sample, with only three lawyers completing the survey, we can conclude that the program developed worked well and was functional, as all the responses received on functionality (Q1, Q2) and usability (Q3, Q4) were positive. Another positive outcome is that two of the three ECHR lawyers responded that they trusted that the justifications for the decisions made were sensible and understandable (Q5) and all three respondents agreed that the information was easy to parse (Q6). All respondents saw the usefulness of our prototype (Q7), with two respondents stating they would use the program as it currently stands, and the other affirming the usefulness but saying that some (as opposed to many) changes are needed. Again all the respondents agreed that technology has a role to play in the legal domain (Q8): one respondent said technology is needed rapidly, while two cautioned that careful development will be needed. The positive responses to Q9 and Q10 were particularly pleasing, since these questions directly concerned the central aims of this exercise: the users all agreed that the questions were suitable for them and that the program would save them time when assessing admissibility. While the majority of the feedback has been positive, it has also highlighted the need for domain experts to be a part of the development process (Q11): although two respondents felt the program reflected all or part of their own process of dealing with admissibility, one felt that only some aspects had been covered. Overall the response to the program is very positive and indicates that it is a sound basis for further developments for our legal decision support tools, giving potential for collaborations between computer scientists and lawyers. Encouraged by these results, we will now extend the study to a larger group of potential users, to ensure that this initial evaluation is reflected in the law community.

7 SUMMARY AND NEXT STEPS

Our aim in this paper has been to conduct a deep dive into the legal analysis required to provide a robust, executable model of Article 6 of the ECHR. We focused on the theory and practice of a particular issue within the the ECHR, namely admissibility of cases. In consultation with our legal expert, we defined an ADF that captures the domain knowledge relevant for the issue of admissibility and transformed our model into an implemented tool that is able to ask questions appropriate to the target user. This is a necessary step to enable academic exercises on legal case-based reasoning to be transformed into usable tools. Our prototype tool implements the back-end reasoning of the underpinning ADF and provides us with solid grounding on the road to development of a tool for use in practice.

The prototype tool has been tested by a small group of the target audience of lawyers. heir response has been very positive, highlighting that lawyers are receptive to tools that are carefully designed and targeted at problems they encounter with their work, all of which paves the way for a fuller evaluation.

The continuation of the work will focus on expanding on the use of the prototype. We are currently organising, with the assistance of our ECHR expert, a field evaluation of our tool by users who work within the ECtHR. This will provide a greater range of feedback in order to adjust our tool as needed to best fit the audience's needs. Once we have a satisfactory tool for use by those assessing applications - which would provide a way of addressing the current significant backlog of unprocessed cases in the ECtHR - we will develop a set of questions for use by applicants themselves. Such a facility should reduce the number of inadmissible applications by enabling applicants to gain a better understanding of what is required to make an admissible application. In this way, access to to the ECtHR could be improved, both by assisting applicants, and by speeding up the decision process for cases submitted.

REFERENCES

[1] L Al-Abdulkarim, K Atkinson, and T Bench-Capon. 2016. A methodology for designing systems to reason with legal cases using ADFs. *AI and Law* 24, 1 (2016), 1–49.
[2] N Aletras, D Tsarapatsanis, D Preoţiuc-Pietro, and V Lampos. 2016. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ Computer Science* 2 (2016), e93.
[3] V Aleven. 1997. *Teaching case-based argumentation through a model and examples.* Ph.D. thesis. University of Pittsburgh.
[4] T Bench-Capon. 1991. Practical legal expert systems: the relation between a formalisation of legislation and expert knowledge. In *Law, Computer Science and Artificial Intelligence*, M Bennun and A Narayanan (Eds.). Ablex, 191–201.
[5] T Bench-Capon, G Robinson, T Routen, and M Sergot. 1987. Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In *Proceedings of the 1st ICAIL*. 190–198.
[6] G Brewka and S Woltran. 2010. Abstract dialectical frameworks. In *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning.* 102–111.
[7] S Brüninghaus and K Ashley. 2003. Predicting outcomes of case based legal arguments. In *Proceedings of the 9th ICAIL*. ACM, 233–242.
[8] I Chalkidis, I Androutsopoulos, and N Aletras. 2019. Neural legal judgment prediction in English. *arXiv preprint arXiv:1906.02059* (2019).
[9] J Collenette, K Atkinson, and T Bench-Capon. 2020. An explainable approach to deducing outcomes in European Court of HumanRights cases using ADFs. In *Proceedings COMMA 2020.* IOS Press, 21–32.
[10] F De Londras and K Dzehtsiarou. 2018. *Great Debates on the European Convention on Human Rights.* Macmillan International Higher Education.
[11] European Court of Human Rights. 2019. Analysis of statistics. (2019). https://www.echr.coe.int/Documents/Stats_analysis_2019_ENG.pdf.
[12] M Grabmair. 2017. Predicting trade secret case outcomes using argument schemes and learned quantitative value effect tradeoffs. In *Proceedings of the 16th ICAIL.* 89–98.
[13] A Kaur and B Bozic. 2019. Convolutional Neural Network-based Automatic Prediction of Judgments of the European Court of Human Rights.. In *27th AIAI Irish Conference on AI and Cognitive Science.* CEUR 2563, 458–469.
[14] M Medvedeva, M Vols, and M Wieling. 2019. Using machine learning to predict decisions of the European Court of Human Rights. *AI and Law* (2019), 1–30.
[15] M Medvedeva, X Xu, M Vols, and M Wieling. 2020. URI SAYS: An Automatic Judgement Prediction System for the European Court of Human Rights.. In *Proceedings of JURIX 2020.* 277–280.
[16] M Sergot, F Sadri, R Kowalski, F Kriwaczek, P Hammond, and H Cory. 1986. The British Nationality Act as a logic program. *Commun. ACM* 29, 5 (1986), 370–386.
[17] D Skalak and E Rissland. 1992. Arguments and cases: An inevitable intertwining. *AI and Law* 1, 1 (1992), 3–44.