

THE RESEARCH UNIVERSITY PRESS

International
Computational Law Journal

VOLUME 2 / ISSUE 6 / CURATED SOURCE EDITION

Flash Talks and Idea Flow Series Kickoff

Dazza Greenwood · Tiemae Roquerre · Megan Ma · Andrew Domzalski · Bryan Wilson

MIT Computational Law Report · Originally published 2021-02-26

Independent reader edition under CC BY 4.0. This attribution cover was added; original source PDF pages follow unchanged. Original authorship and source identity are retained. MIT, Stanford University, Stanford Law School and CodeX have not sponsored or endorsed this republication.

Source, license and version evidence accompany the online article.

MIT Computational Law Report • Media

Flash Talks and Idea Flow Series Kickoff

Dazza Greenwood, Tiemae Roquerre, Megan Ma, Andrew Domzalski, Bryan Wilson

Published on: May 14, 2021

License: Creative Commons Attribution 4.0 International License (CC-BY 4.0)

This episode of Idea Flow includes an introduction to the series and flash talks from Megan Ma and Tiemae Roquerre.

Visit the web version of this article to view interactive content.

Idea Flow - Episode 1

The Legislative Recipe: Syntax for Machine-Readable Legislation

Speaking on "The Legislative Recipe: Syntax for Machine-Readable Legislation" Megan poses the following remarks:

The prospect of machine-readable legislation is both terrifying and thrilling. This renewed popularity is owed to the Rules as Code initiative. The fervor around Rules as Code was accelerated by the recent OECD Observatory of Public Sector Innovation Report titled, "Cracking the Code."

This report articulates how machine-consumable, defined as machines understanding and actioning rules consistently, "reduces the need for individual interpretation and translation" and "helps ensure the implementation better matches the original intent." This methodology enables the government to produce logic expressed as a conceptual model – in effect, a blueprint of the legislation.

So, what is the attraction and what are its limits?

I frequently turn to this example. Layman E. Allen lamented about ambiguity in legal drafting owed to syntactic uncertainties. In a fascinating study, he deconstructs an American patent statute and notices immediately the complexity with the word 'unless.' He asks whether the inclusion of 'unless' asserts a unidirectional or a bidirectional condition. That is, does the clause mean (a) if not x then y; or (b) if not x then y and if x then not y?

Though nuanced, Allen exposes an ambiguity that muddies the legal force of the statute. An interpretation of 'unless' as a bidirectional condition raises the question of what "not y" would mean. In this particular case, this could affect whether exceptions are possible in determining patent eligibility. In short, for Allen, legislative language must have a clear structure.

These ideas are not new. The ancestry dates back to twelfth century logicians reflecting on the use mathematically precise forms of writing. In the mid-1930s, German philosopher, Rudolf Carnap, reflected on a logical syntax for language. His argument is that logic may be revealed through the syntactic structure of sentences. He suggests that the imperfections of natural language point instead to an artificially constructed symbolic language to enable increased precision. Simply put, it is treating language as a calculus.

More recently, Stephen Wolfram made a similar argument. Simplification, he states, could occur through the formulation of a symbolic discourse language. That is, if the "poetry" of natural language could be "crushed" out, one could arrive at legal language that is entirely precise.

Machine-readability appears then to bridge the desire for precision with the inherent logic and ruleness of specific aspects of the law. In other words, a potential recipe to resolve the complexity of legalese. However, if a new symbolic language, like code, effectively enforces a controlled grammar, what are its implications as it moves across the legal ecosystem (in particular, its interactions between various legal texts)?

Machine-readable legislation may, therefore, be regarded as a product that evolved out of the relationship between syntax, structure, and interpretation. But at the core, it boils down to one question: what should be the role of machine-readable legislation? Is it simply a 'coded' version of legislation? (one possible interpretation); or is it a parallel draft of the legislation (one that is legally authoritative)? Or, is it a domain model of regulation from which third parties derive versions (open-source code)?

These three scenarios have their own sets of implications. And only in answering this question, would a fruitful assessment of how logic syntax and symbolic language, found in machine- readable legislation, are capable of representing legal knowledge.

A Discussion on Algorithmic Sentencing

In the second flash talk, Tiemae Roquerre, provides "A Discussion on Algorithmic Sentencing" and presents the following challenge:

Recently, as countries like Estonia and Singapore have started experimenting with algorithmic sentencing in small claims courts, it's become more important than ever before to think about shifts that algorithmic sentencing may impose on existing judicial processes — particularly the trust and standards that apply to the role of a judge.

Because the specifics of Justice systems around the world vary so much, for the sake of simplicity, we'll keep this discussion to the US justice system.

In the US, it's currently pretty well established that our justice system is riddled with ingrained biases and inequities: people of color are not only overrepresented as defendants in our criminal justice system, but they also receive longer sentences than white defendants. And these injustices seem to be the byproduct of human biases and prejudices in sentencing by judges, which algorithms could certainly avoid.

Many judges today would concede that a mere spreadsheet providing data on past sentencing decisions could help them make more objective decisions. So in this vein, it seems like carefully coded sentencing programs could skirt situations like when an Ohio judge went against the recommendations of both the defense council and the state prosecutor to condemn a 55-year-old woman, who was a first-time nonviolent offender, to 65 years in prison — for petty theft. Or when a man was condemned to a life in prison for merely attempting to steal a set of hedge shears.

It's true that a non-trivial objection to algorithmic sentencing is that data used to program the code is often incomplete or incorrect, biasing outcomes. But assuming this can be remedied, we must ask if society would even be amenable to algorithmic justice.

In the United States today, judges are expected to be the arbiters of Justice. The Code of Conduct for United States Judges states as its first canon: "A Judge Should Uphold the Integrity and Independence of the Judiciary. Further they should not only maintain and enforce high standards of conduct, but they should personally observe those standards, so that the integrity and independence of the judiciary may be preserved." Are robot judges able to personally observe anything? And could they in turn fulfill this standard?

Under the Social contract theory, the concept of Justice exists because of collectively negotiated human belief. Does the US justice system work in part because Americans believe in the idea of human judges as arbiters of Justice. Or is it unnecessary?

It is clear that the advent of algorithmic sentencing is calling into question the role of human judges as referees of Justice, So in light of some of these considerations, what are the pros and cons of both human judges and algorithmic sentencing? And what should the best path forward look like?