The Legacy of Hammurabi

Michael Genesereth
CodeX: The Center for Computers and Law
Stanford University

1. Introduction

Around 1750 BC, Hammurabi had the laws of the land encoded in written form (literally cast in stone) so that citizens could know what was expected of them and what would happen if they violated those expectations. Thus began a tradition that has lasted through millennia. One that has served us well. At the same time, a tradition that is not without its problems.

Today, we live in a complex regulatory environment. As citizens, we are subject to governmental regulations from multiple jurisdictions - federal, state, and local. As social beings, we are bound by contracts we make with others.

The sheer number and size of regulations can be daunting. We may all agree on a few general principles; but, at the same time, we may disagree on how those principles apply in specific situations. The Declaration of Independence is an important document in American history. It outlines the principles on which the country is based in just 1322 words. "We hold these truths to be self-evident - that all men are created equal. And so forth." By contrast, the regulations on the sale of cabbages run to 26,911 words. That's not bad writing. The number and size of the regulations is essential to deal with details and special cases.

Complicating the situation is the complexity of these regulations. Even small regulations can be very complex. Moreover, once regulations are created, complexity often increases as the regulations are changed and then changed again. An example from my homeownerâ€™s insurance policy is shown here. This sort of thing is typical in insurance contracts. There is a statement on page 112 stating that the coverage on page 32 does not apply when various conditions exist.â€ The upshot is a policy that is difficult for most people to understand without specialized legal knowledge and a substantial amount of study.

To make matters worse, regulations are not always well coordinated, arising, as they do, in different jurisdictions. Sometimes, there are gaps, leaving important cases uncovered. More often, regulations overlap other regulations and in some instances are inconsistent with each other.

These problems make it difficult for affected individuals to find and comply with applicable regulations. The result is occasional lack of compliance, widespread inefficiency, and frequent disenchantment with the regulatory system.

This is a failure of our legal system. One of the functions of the law is to help individuals predict the consequences of their actions. If we do not know what the law is, the law does not serve this function; and, as many people have observed, the law today is far too complex for people to understand fully.

2. Computational Law

Fortunately, these problems are not insurmountable. As a computer scientist, I see these as information processing problems. As such, I believe that they can be mitigated by information technology. What is needed is appropriate "legal technology" - information technology applied to laws.

One step in this direction has already been taken. Today, the text of most legal documents in many countries (including statutes, regulations, cases, analysis) is available online. In some cases, the information is adorned with "semantic" tags / keywords to help in search. The good news is that these documents can be found using general search services, such as Google, or using services that specialize in legal information, e.g. those provided by companies like Westlaw and LexisNexis. Unfortunately, the quality of such search is limited; they often return too many documents and sometimes fail to find relevant information. More importantly, there is no automation; a specialist must still be there to read the documents and apply them to individual cases.

An alternative, the subject of my remarks today, is an extreme form of legal technology known as Computational Law. Computational Law is that branch of legal informatics concerned with the automation of legal reasoning. Not just legal search but legal analysis. Answers, not documents!

Intuitâ€™s Turbotax is an oft-cited example of a Complaw application. Millions use it each year to prepare their tax returns. Based on values supplied by its user, it automatically computes the userâ€™s tax obligations and fills in the appropriate tax forms. Extensions provide users with support in financial planning. In principle, it can be used by regulators in exploring the consequences of hypothetical changes to the tax code.

However, taxes are not the only application of Computational Law. There are many other areas of the law that are amenable to similar treatment. Portico is a prototype of a system developed at Symbium for assisting architects and homeowners in formulating architectural designs that comply with planning codes and building codes. Analogous systems can be built other areas areas - e.g. management of child support, immigration aids, and building inspections.

There are also applications that do not involve governmental laws. The rules and regulations can just as well be the terms of contracts (e.g. insurance covenants, delivery schedules, real estate transactions, financial agreements). There are applications in enterprise managements. The rules can be the policies of universities (e.g. the academic program sheet pictured here) or the business rules of commercial corporations (e.g. travel expense reimbursement, product configuration worksheets, and pricing rules). And there are game-playing systems - where the laws" are the rules of games.

Over the years, various companies have sprung up to meet the need here. See the CodeX techindex for 1350 companies in legal tech, including dozens doing work in Complaw.

3. Technology

Complaw systems are computer systems and as such can be built in much the same way as other computer systems. However, complaw systems are about the Law; and, as such, there are certain opportunities and challenges in building complaw systems that do not exist in other applications of computer technology. In this section, we talk about three issues that arise in this regard - the representation of laws, the application and analysis of laws, and the creation ad modification of laws.

In traditional programming, systems take the form of programs written in languages like C and Java. The programs in such cases are typically imperative - they tell the system what to do step by step. And programs are typically task-specific - each is designed to accomplish a specific task. ... However, this is not the architecture envisioned by most AI researchers.

The alternative approach is declarative programming. In this approach, regulation-specific programs are replaced by regulation-independent, and the regulations are supplied as data just like facts. In this case, the â€œprogramâ€ is declarative, and the program itself is independent of any particular task.

One advantage of separating representation and reasoning in this way is that a single general legal reasoning system can be used multiple times, for different jurisdictions and for different combinations of jurisdictions.

The dual of this is also true. Once a set of regulations is encoded formally, it can be supplied as input to different legal reasoning engines for different purposes, e.g. to check compliance, to plan for compliance, to detect inconsistencies or redundancies, and so forth.

The vast majority of rules and regulations are encoded in natural language. So it would be convenient if we could just use those.

Unfortunately, this is not that easy. We not today have computer technology that is adequate to the task. There has been notable progress (as evidenced by the popularity of systems like Siri and Alexa and Google Translate), but today's systems still make too many errors to be used in autonomous Complaw applications.

But the problems are not just technological. Languages like English are very expressive but they have fundamental ambiguities and complexities that limit their suitability.

One problem is that natural language is often grammatically ambiguous. Consider this sentence from a popular textbook on Symbolic Logic: There's a girl in the room with a telescope. Here we see two possible meanings of this sentence. Does the sentence mean that there is a girl in a room containing a telescope? Or does it mean that there is a girl in the room and she is holding a telescope?

A more insidious problem is that natural language includes syntax that can be misread, leading to bad conclusions. Let's say we are told that champagne is better than beer and that beer is better than soda. From these premises, we might conclude that champagne is better than soda. This is an example of transitivity. If X is better than Y and Y is better than Z, then X is better than Z. Now, consider what happens when we apply the same transitivity rule in the case illustrated here. Bad dessert is better than nothing, and nothing is better than good dessert. A NLU system might use transitivity to conclude that bad dessert is better than good dessert. The form of the argument is the same as before, but the conclusion is somewhat less believable. The problem in this case is that the use of the word "nothing" here is syntactically similar to the use of beer in the preceding example, but in English it means something entirely different.

The alternative is to use a formal language for encoding information - one with well-defined syntax and clear semantics. The benefit of formal languages is that we can engineer them so that they are not ambiguous and we can avoid complexities that are likely to lead to bad conclusions.

The most popular approach to building Computational Law systems using formal languages is based on Computational Logic. For example, in logic programming, information takes form of

"rules" written in the language of symbolic logic. Such languages allows us to define real world concepts. They also allow us to write definitions for legality and illegality. In this case, the rules come from a corporate policy handbook, but the approach applies equally well to governmental rules.

One thing that makes Computational Logic useful is the availability of automated reasoning programs that can be used to apply rules to facts (in the form of structured data) to derive logical conclusions. They can be used to generate legal plans. And they can be used to detect inconsistencies, overlaps, and gaps in regulations.

Laws expressed within such languages can be translated to other languages. Natural language generation is easier than natural language understanding and the results are more reliable.

The development of formal representations can be made easier though the use of interactive development environments tailored to the codification of laws - programs that can accept inputs in natural language and pose possible translations, programs that can check for inconsistencies and incompleteness, programs that can pose and handle hypothetical cases to check that encodinigs are correct.

Unfortunately, things are not perfect. Not due to technological deficiencies but because the laws themselves have problems.

For example, they may be incomplete. A technical problem, familiar to many individual with legal training, is due to the open texture of natural language. Consider a municipal regulation stating â€œNo vehicles in the parkâ€. On first blush this is fine, but it is really quite problematic. Just what constitutes a vehicle? Is a bicycle a vehicle? What about a skateboard? How about roller skates? What about a baby stroller? A horse? A repair vehicle? For that matter, what is the park? At what altitude does it end? If a helicopter hovers at 10 feet, is that a violation? What if it flies over at 100 feet?

To make matters worse, regulations are not always well coordinated, arising, as they do, in different settings for different purposes. Sometimes, there are gaps, leaving important cases uncovered. More often, regulations overlap other regulations and in some instances are inconsistent with each other.

There are good reasons that such problems arise. There can be changes in society - e.g. the move from agrarian economy to industrial economy, the move from cities to suburbs. In some cases, the rules are stretched by changes in technology, e.g. reusable rockets, autonomous cars, gene editing - developments that strain the rationale for the rules on the books.

In some cases, rules are intentionally oversimplified - to save the effort of enumerating all of the exceptions and to make the rules easier for people to understand and apply.

In some cases, rules are left intentionally ambiguous in order to make it possible for politicians and regulators to compromise, effectively "kicking the can" down the road for other regulators and courts to deal with..

The good news is that, in common law countries, like the US, we have a mechanism for dealing with such cases. We have judges and courts to hear specific cases and to further refine laws and regulations. And we have appeals courts and in the US, a supreme court to ensure that they do it right.

The problem with this is that it complicates the task automating legal analysis. The good news is

that there are some technological ideas for dealing with this part of the problem.

In some cases, rules are intentionally oversimplified - to save the effort of enumerating all of the exceptions and to make the rules easier for people to understand and apply.

One possibility for dealing with cases like this that has been proposed is automated adjudication - not just applying logical rules but rather modifying or elaborating rules to handle specific cases.

Unfortunately, this needs something more than logical reasoning / deduction. In the words of Edwina Rissland: "Law is not a matter of simply applying rules to facts via modus ponens", and when regarding the broad application of AI techniques to law, this is certainly true. The rules that apply to a real-world situation, as well as even the facts themselves, may be open to interpretation, and many legal decisions are made through case-based reasoning, bypassing explicit reasoning about laws and statutes. The general problem of open texture when interpreting rules, along with the parallel problem of running out of rules to apply when resolving terms, presents significant obstacles to implementable automated rule-based reasoning [17]. Also, in many legal domains, the facts of a situation themselves may be unclear or incomplete: human intervention and interpretation is necessary to make these facts available to a legal reasoning system so that it can even apply the rules. This further adversely affects usability and any notion of correctness. To combat these shortcomings, some rule-based systems have been hybridized with case-based systems, or augmented with meta-rules or with nonmonotonic, defeasible reasoning techniques, in order to make them more suitable for general applications in law.

Another approach is to apply statistical reasoning to past decisions in the hopes of producing predictions of how courts would decide new cases. The work of Kats, Bommarito, and Blackman suggests that this more than a pipe dream. Their work on predicting the outcomes of Supreme Court decisions outstrips that of humans given the same information, suggesting that it might be possible to deploy technology to give good estimates of judicial outcomes for those cases where rules are not completely adequate.

Finally, there is discussion of the possibility of introducing automation into the regulatory process itself, possible generating rules and regulations that cover more cases from the outset and thereby obviate the need for adjudication after the fact.

Despite these problems, technologies of this sort are in use today in business. Virtually all large corporations today utilize enterprise management software applications to run the operations of their businesses, such as accounting/finance, human capital management, supply chain & manufacturing, etc. Pricing rules, privacy rules, expense reimbursement, and so forth. The development of such software and services has led to sizable businesses for companies like SAP, Oracle and IBM.

The good news is that it is not used as extensively with the largest on the planet. Which is? Not energy. Not healthcare. Not telecommunications. It is government. My position is that these same technologies can and should be applied to the public sector, except with governmental rules and regulations in place of business rules, in areas like those mentioned earlier.

4. Embedded Law

The potential for deployment of Complaw applications is substantial due to technological developments like the Internet, mobile systems (such as smart phones and smart watches), and the emergence of autonomous systems (such as self-driving cars and robots). All of which allow us to

make automated legal analysis tools available to CITIZENS in their daily lives. Which brings me to the main point of my presentation today - the cop in the backseat.

Suppose that we had the benefit of a friendly policeman in the backseat of our car whenever we drove around (or perhaps an equivalent computer built into the dash panel of our car). The car can and should offer regulatory advice as we drive around - telling us speed limits, which roads are one-way, where U-turns are legal and illegal, where and when we can park, and so forth. The Cop in the Backseat. But a friendly cop rather than a punitive one.

Capabilities like this already exist to limited extent in aviation, where displays like this one provide feedback on restricted areas and areas with special requirements (these concentric circles). On the Internet when we are deciding whether to buy that drug from Canada or ship that alcohol to Virginia.

The ubiquity of computer technology makes lots of things possible. You are walking through the woods of Massachusetts and you see an attractive flower. You take a photo with your iPhone. Your plant app identifies it as a type of orchid and lets you know. At the same time, your legal app tells that, no, you may not pick it.

In some locales, speed limits are based on location, e.g. the lane of travel. Speed limits sometimes based on time of day. Speed Limits could be based on type of vehicle. Speed limits based on personal characteristics. The FAA already does this.

Technology is making it possible for us to enforce laws in ways that were not previously feasible. Automated reporting and billing is one potential of technology. Red Light cameras are examples.

So far we have been talking about a cop that is friendly rather than punitive one. (Maybe we should instead consider the possibility of a "Lawyer in the Backseat" or a "Driving Instructor in the Backseat".)

The alternative is a cop with the power to ding us for violations of the law when we do not take his/her/its advice. In the case of a computerized policeman with an internet connection, we could imagine the policemen immediately reporting the violation to the DMV (Department of Motor to Vehicles).

Insurance companies already make devices that track and report driving performance, allowing conservative drivers ti benefit from lower rates and increasing the premiums for more aggressive drivers.

Taking this one step further, we can imagine cars showing the results of such reporting to their drivers as well other performance factors. The cars could display not just actual speeds, but also speed limits, DMV fine balances, insurance premiums, and so forth.

It would be interesting to see the effects of such reporting on drivers. Would they drive more conservatively when they see their bills mounting every time they exceed the speed limit?

It seems clear that there are positive features to conditional laws like these. They promote safety while enhancing efficiency. At the same time, there are concerns, e.g. whether it is equitable to discriminate on the basis of personal characteristic, whether automatic DMV reporting compromises our right to privacy, and so forth.

When I mention these possibilities, I often hear â€œNo way, no how. It will never happen.â€ Maybe so. The question is whether it should. Is it a good idea or a bad idea? If it is a good idea, how can we help to make it happen? And, if it is a bad idea, how can we prevent it from happening?

5. Conclusion

There are those who think of complaw as just another tool to support the work of lawyers, as some sort of smart typewriter with no significant legal value. I am not one of these. I believe that Computational Law has significant implications for our legal system.

I began my talk by remarking on the size and complexity of regulations and outlining some of the pitfalls - lack of compliance, widespread inefficiency, and frequent disenchantment with the regulatory system.

I have argued that Complaw has the potential to mitigate these problems - enabling automated compliance checking, legal planning, regulatory analysis, and so forth. But its value does not stop there.

It can also lead to better laws. The fact is that there is virtue in complexity. We need complex rules to cover all of the cases rather than resorting to one-size-fits-all rules. But there is also harm in complexity. Complexity is the enemy of understanding. The use of computational tools allows us to reap the benefits of complexity without the costs. It allows us to make BETTER LAWS. It allows us to make a BETTER LEGAL SYSTEM.

And there is a broader implication as well. Complaw technology has the potential for democratizing the law. It takes law out of the courtroom and the law office and makes it available to people who are not legal professionals. It makes it possible to embed the law in the real world, making it available to ordinary decision makers at the point of decision, when they are about to act or planning how to act. It can alert people to their obligations; it can help people understand their rights and privileges; it can help people get their due from the government, from insurance companies; and so forth.

Hammurabi began a tradition that has lasted through millennia. Since then, it has been the norm to encode rules in written form and disseminate first via books and more recently via the Internet. However, just writing things down is not enough when the laws are voluminous and difficult to understand. In a sense, Computational Law is the ultimate step in a progression that began millennia ago. And it is the basis for a legal system that works for everyone. This is the real Legacy of Hammurabi.