
STATES OF JERSEY



COMPUTER READABLE LEGISLATION PROJECT

**Presented to the States on 10th February 2026
by the Privileges and Procedures Committee**

STATES GREFFE

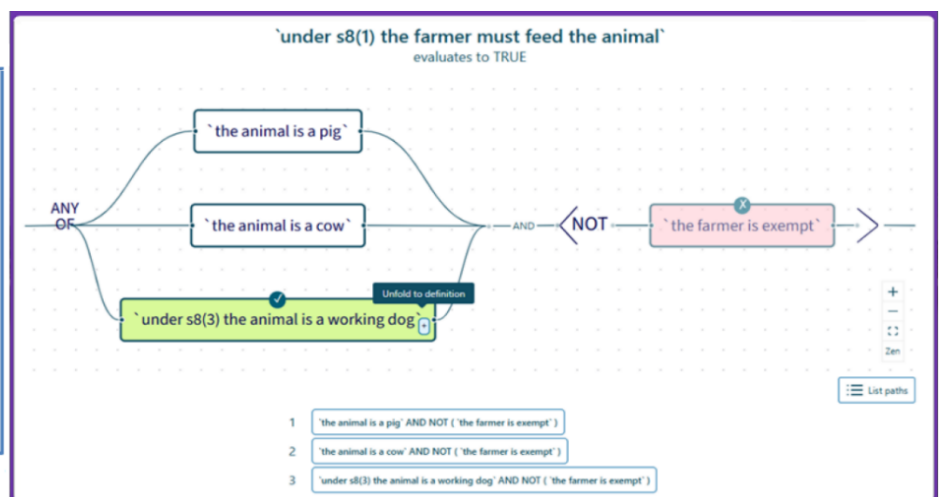


Computer-Readable Legislation Project

Report to the Privileges and Procedures Committee, the
States Assembly and the Jersey Legal Information Board

January 2026

s8(1) A farmer must feed an animal if -
(a) the animal is -
(i) a livestock animal; or
(ii) a working dog; and
(b) the farmer is not exempt.
(2) In this Act "livestock animal" means -
(a) a pig; or
(b) a cow.
(3) In subsection (1) "working dog"
means a dog that -
(a) works as a sheepdog, and
(b) is registered.



- (4) The Commissioner must publish and maintain guidance on the determination of whether an entity meets the charity test.
- (5) Any person, in determining whether an entity meets the charity test, must have regard to the guidance, and the Commissioner, the tribunal, a registered charity and a governor of a registered charity must have regard to the guidance when performing any other of their functions under this Law to which the guidance is relevant.
- (6) Before issuing or amending guidance the Commissioner must –
 - (a) consult –
 - (i) any persons appearing to the Commissioner to be representative of charities or bodies with charitable purposes.
 - (ii) the Minister, and
 - (iii) such other persons as the Commissioner considers appropriate; and
 - (b) publish a report on the Commissioner's views on the results of the consultation, and the Commissioner's decision on the guidance in the light of those results.
- (7) Within a reasonable time after issuing or amending guidance the Commissioner must provide to the Minister a copy of the issued or amended guidance and the report

Summary

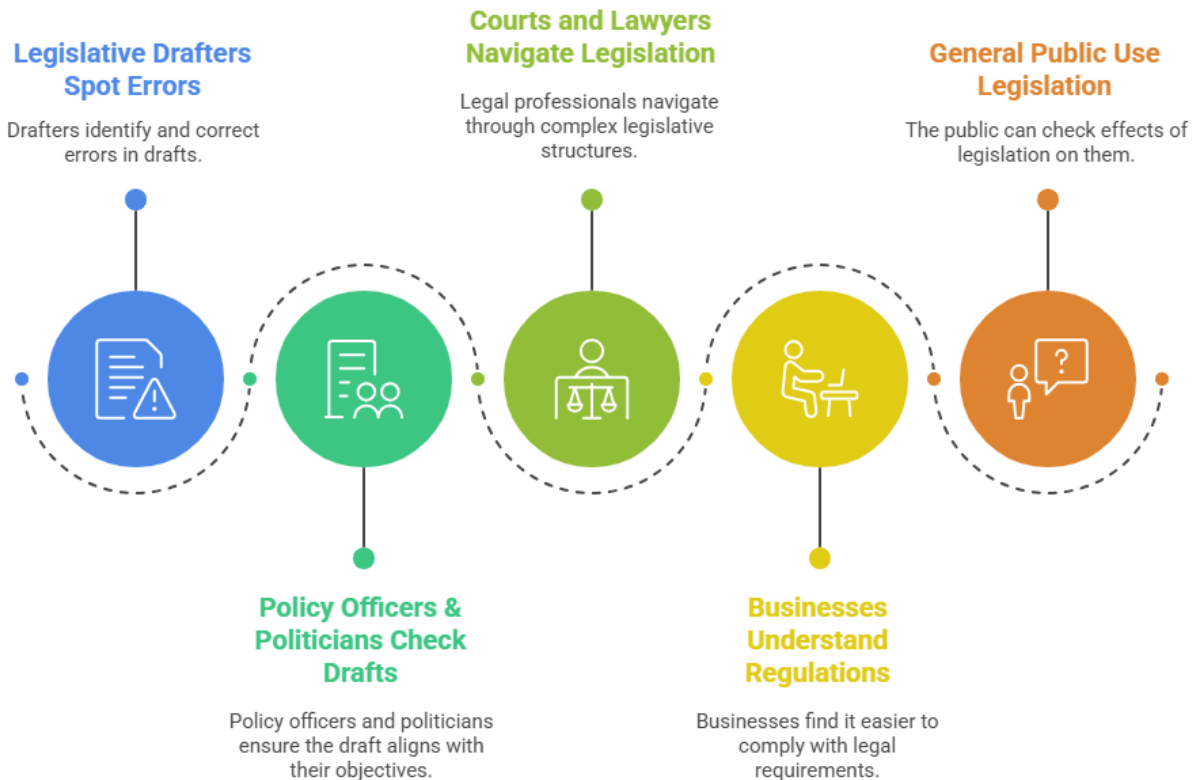
Introduction to this report

This report sets out the work and achievements to date of the [Computer-Readable Legislation Project](#) run by [Jersey's Legislative Drafting Office](#). It sets out the aims of the project, its global impact and its organisation. It details the main results of the project – our logic visualiser, ways to use AI, our definitions tool and ideas for a drafting tool, our drafting guidance, our illustrative tool for drafting instructions, our set of example provisions, and our publications. It ends with some notes on the [background](#) to our approach (including a guide to [key concepts](#)). A full contents page follows after this summary, and the body of the report gives fuller details, explanations and links.

What the Computer-Readable Legislation Project does

Our project is looking for ways in which legislative drafters can ensure that computers can read the logical structure of the legislation we draft. That has benefits in several ways –

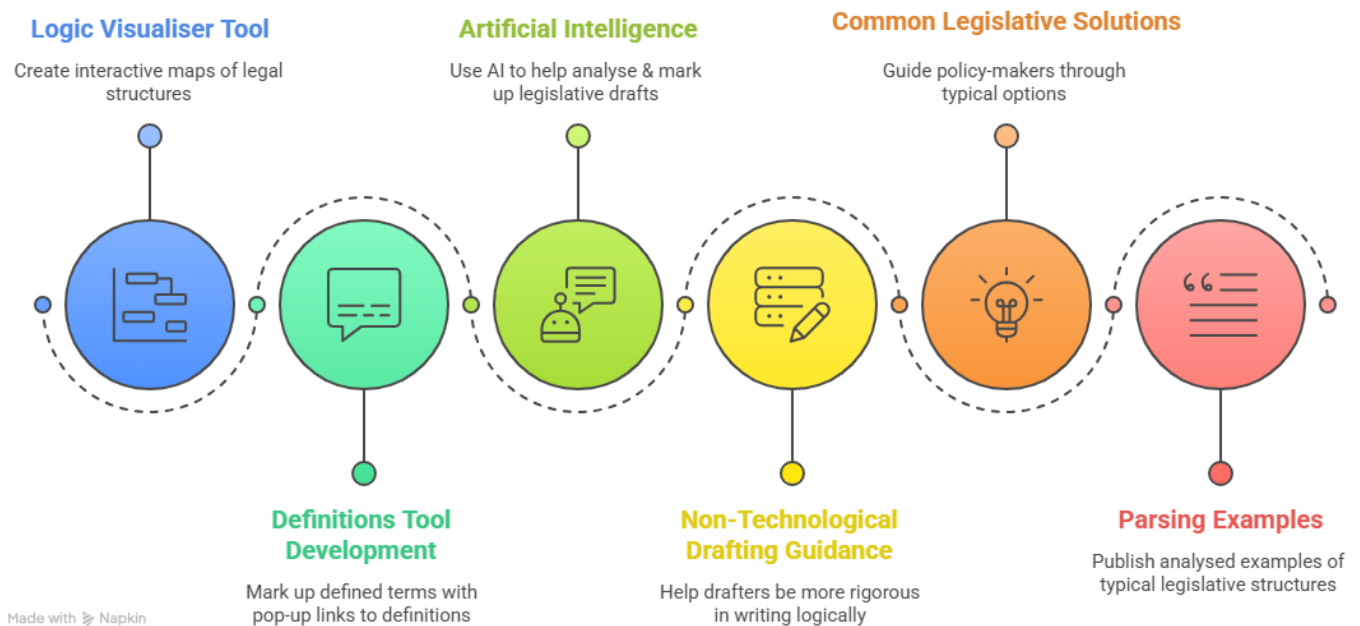
- helping legislative drafters to spot and fix errors in a draft before it is made into law,
- helping policy officers and politicians to check that the draft does what they want it to do, and
- helping courts, lawyers, businesses and the general public to navigate their way through reading the resulting legislation (when the legislation is complex because it reflects the complexity of modern life and government).



The project's global impact

The project's results have been shared widely among legislative drafting offices across the Commonwealth, and among technology experts working in this field worldwide. That has brought direct benefits of collaboration. By achieving this level of international recognition, the project has also contributed to enhancing Jersey's reputation as a world-leading innovator in both the legal and digital fields.

The project's main results so far



Our focus has been to ensure that what we produce can be used by legislative drafters without having to download any software, pay any fee or learn to code. We are particularly keen on illustrative, interactive demonstrations that help legislative drafters (and developers) to understand the sorts of tools they could have and the sorts of benefits those could bring.

- **Logic visualiser** – This is a free, web-based tool that legislative drafters can use to generate an interactive map of the logical structure of their draft. They take an if-then version (in English text drawn from the draft) of the legal effect and conditions in their draft provision, and automatically turn that into a basic version of a coding language called “[L4](#)”. The drafter copies the L4 code into the visualiser which then automatically generates the interactive map of the conditions and effect. The drafter, or another user like the policy officer, can then click on boxes containing the conditions to indicate whether they are met or not. The visualiser then automatically updates the map to show whether the legal effect applies or not. The user can give different answers to check that various fact scenarios all have the desired results when the draft provision is applied to them.
- **Definitions tool** – The Legislative Drafting Office has developed a tool that drafters are currently using to mark up all defined terms with pop-ups that show the definition and give a link back to it. That includes definitions in the [Interpretation \(Jersey\) Law 1954](#) and in a Law under which Regulations or an Order are made, which readers may otherwise be unaware of. We have had positive feedback from policy officers where we have applied it to active drafts.

- We are now testing the system to see whether it can be extended to legislative drafts in Assembly propositions and eventually (we hope) to new legislation published on Jersey’s legislation website (jerseylaw.je).
- The system is built on new identifiers that are automatically applied to each paragraph of text. The identifiers reflect the structure and are both human-readable and computer-readable. They are likely to enable us to develop further tools and features in future.
- We have also produced an interactive mock-up of a more wide-ranging use of pop-ups and links. We have published additional ideas about how legislative drafters could have writing tools that match the usefulness to coders of their “integrated development environments” (or “IDEs”) – perhaps an integrated legislative drafting environment (or “ILDE”).
- **Artificial Intelligence** – We have found ways to use generative AI (particularly chatbots using Large Language Models, or “LLMs”) in ways that play to its strengths but avoid relying on it for jobs that it is not appropriate for. While LLMs continue to generate the most probable text, “hallucinations” will remain a feature, making them unreliable for use on their own in working out how legislation applies to given facts. Our work on AI includes –
 - We took the initiative, working with the Law Officers’ Department, the States Greffe and the Jersey Legal Information Board, to produce an open copyright licence for our legislation, so AI chatbot developers can legally use it. JLJB then set up an access point (an application programming interface or “API”), to avoid problems caused by AI developers scraping the legislation website. We then produced notes for developers of chatbots to help them see how Jersey legislation works and where LLMs can fall down.
 - We published our take on how chatbots can be used in a hybrid (or neuro-symbolic) “AI sandwich” as an interface for a human to use a reasoning system, and have contributed to the work on using L4 for that.
 - We worked with Digital Jersey on a project to use Google Cloud Document AI to identify elements in legislation such as offences. That taught us useful lessons on techniques in training AI but also identified past poor drafting practices on offences which we have now improved. We have also encouraged others to apply machine learning to our legislation, and other forms of AI beyond LLMs.
 - We helped set up the AI working group of the Commonwealth Association of Legislative Counsel, and have liaised with the Jersey AI Forum and the government’s Digital Services.
- **Non-technological drafting guidance** – We have produced guidance for Jersey’s legislative drafters on how to improve our drafting of the English text of legislation, based on insights from the project’s work on logic and computing.
 - For Jersey’s Legislative Drafting Office, we have run multiple training sessions with the drafters and other members of the team, written guidance tailored to Jersey, and created pages on Saver (the office’s internal knowledge management wiki).

- What we have learnt is adaptable to other Commonwealth drafting offices, so we have also run training and demonstration sessions at conferences of the Commonwealth Association of Legislative Counsel, and written articles for their Loophole journal and items for their Newsletter.
- These improvements do not depend on any particular technology being developed. Instead they are about things like – how we use “must”, “must not” and “may”; checking what is meant to happen if conditions are not met, obligations are breached, or powers are exercised; checking tenses of verbs in conditions; avoiding syntactic ambiguity & explaining semantic vagueness; and being clear about elements like offences, obligations, prohibitions, permissions, powers, vires, definitions, and other constitutive provisions.
- **[Common Legislative Solutions in Docassemble](#)** – The Legislative Drafting Office has been working on a Jersey adaptation of “Common Legislative Solutions” (a set of questions and guidance for policymakers thinking about what legislative provisions they might need). The office set up a “coding club”, supported by the project, which built on previous work with Docassemble.
- **[Parsing examples and other publications](#)** – We have published our set of imaginary and real legislative provisions that illustrate typical structures that need to be captured for computer-readable drafting. We have also published illustrations of different approaches that can be taken to capturing those structures, using L4 coding, if-then statements and Excel spreadsheets, as well as other freely available technology (such as QnA and Mermaid). More broadly, we have made all our work freely accessible in a repository on OSF (at the Center for Open Science), and we have disseminated our findings through published articles, conference presentations, videos, newsletters and appropriate social media.

How the project is organised

The Computer-Readable Legislation Project is run by Jersey’s Legislative Drafting Office. It had dedicated funding from the Government of Jersey’s budget for the States Greffe in 2023 and 2024, and from the Jersey Legal Information Board in the first half of 2025. The project built on work that the Legislative Drafting Office had already been doing before that funding. The project is now carrying on, without dedicated funding, as part of the work of the Legislative Drafting Office, aided by volunteers who have been involved in the project so far. This work fits with the history of the wider work of the Legislative Drafting Office over many years in using new technology to improve access to Jersey legislation (for example in immediately publishing updated versions of all Covid legislation every time it was amended, unlike most other countries).

Contents

1	Introduction.....	7
1.1	What the project does	7
1.2	Global and local impact.....	7
1.2.1	The project's impact in Jersey	7
1.2.2	The project's global impact	8
1.3	The rest of this report.....	8
2	Logic visualiser – interactive map of logical structure of a legislative provision.....	9
2.1	Using L4 in an interactive logic visualiser of conditions for an effect.....	9
2.2	Using L4 more fully	10
3	Tools to mark up definitions and help drafting generally	12
3.1	Pop-ups and links for definitions – a tool in the testing stage.....	12
3.1.1	The definitions tool.....	12
3.1.2	Paragraph IDs	13
3.1.3	Manual markup during drafting, not later automated markup	13
3.2	Groundwork for a drafting tool.....	15
4	AI – artificial intelligence	17
4.1	Introduction	17
4.2	Enabling AI systems to ingest our statute book.....	18
4.2.1	Licence and API.....	18
4.2.2	Notes for developers of chatbots for Jersey legislation.....	19
4.3	Google Cloud AI with Digital Jersey	20
4.4	Making an AI sandwich with L4 as the reasoner filling	21
4.5	Local and global.....	22
5	Non-technological drafting guidance.....	23
5.1	Guidance and training.....	23
5.2	Types of non-technological drafting improvements	23
5.3	An Interpretation Law for the digital age.....	24
6	Docassemble demo with “Common Legislative Solutions”	25
6.1	Background.....	25
6.2	The mock-up of an app with if-then version & logic map.....	26
7	Making our work freely accessible with global impact	28
7.1	Working in the open	28
7.2	Parsing examples published	30
8	Background	31
8.1	What is legislative drafting?	31
8.2	“Rules as Code”	32
8.3	Guided understanding, rather than automated decision-making (ADM)....	33
8.4	Improvements in new legislation, rather than trying to go back over the existing statute book	34
8.5	A note for coders on logic programming.....	35
8.6	Key concepts and agencies in this field	36
9	Conclusion.....	37

1 Introduction

This report details the achievements so far of a ground-breaking project by Jersey’s [Legislative Drafting Office](#) (part of the States Greffe), building on the Office’s previous work in the field of computer-readable legislation.

1.1 What the project does

The [Computer-Readable Legislation Project](#), run by [Jersey’s Legislative Drafting Office](#), is looking for ways in which legislative drafters can ensure that computers can read the logical structure of the legislation we draft. That has benefits in several ways –

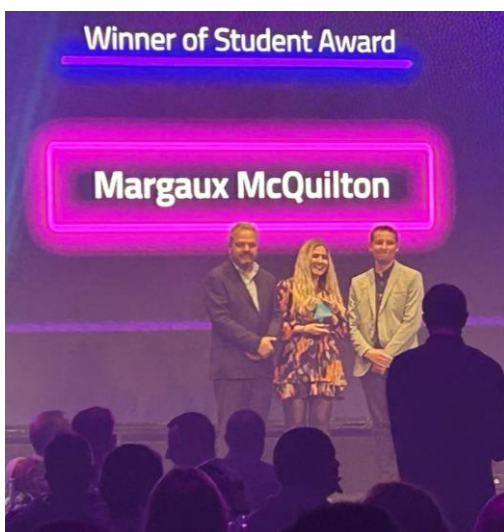
- helping legislative drafters to spot and fix errors in a draft before it is made into law,
- helping policy officers and politicians to check that the draft does what they want it to do, and
- helping courts, lawyers, businesses and the general public to navigate their way through reading the resulting legislation (when the legislation is complex because it reflects the complexity of modern life and government).

Although AI can try to answer questions by reading the English text of our legislation, it will perform much better and be more reliable if we can give it structured data about the legislation and make the logical structure of the legislation computer-readable. Otherwise chatbots will continue to struggle with finding and applying definitions, following cross-references and applying the interpretive principles that legislation relies on. If we can make all of these easier to follow using technology, we will help human readers of the English text as well as AI readers.

1.2 Global and local impact

1.2.1 The project’s impact in Jersey

The project has led to improvements in legislative drafting practices in Jersey and will result in tools that make Jersey legislation easier to read, as set out below.



But we are also proud to have helped to encourage the next generation of lawyers in Jersey to embrace the possibilities offered by technology.

We took on a student from [Jersey’s Institute of Law](#) as assistant. She won an award for her work with us from Digital Jersey, and is returning to our team as one of our volunteers for future work.

We also worked with Jersey’s Institute of Law on their teaching of the University of London’s external LLB module “Legal System and Method”. In each year of the project we have helped the Jersey students by producing a colour-coded analysis of the sections

from an Act that are then used in that year’s London exams. That has made it easier for the students to find their way around the inter-relationships between the provisions.

We also employed an intern on the Government of Jersey summer internship scheme. She now has a full post in policy for the Government of Jersey where she brings a greater understanding of how legislation works, how it is drafted and how technology could improve the process of turning policy into implemented legislation.

We took the initiative to work with the [Institute of Law](#), [Highlands College](#) and [Digital Jersey](#) to set up and run a successful event bringing law students and computing students together. They learnt from each other and the project by working on ways in which computers could capture the logical structure of legislation, in a “mini-hackathon”.

1.2.2 The project’s global impact

The project’s results have been shared widely among legislative drafting offices across the Commonwealth, and among technology experts working in this field worldwide. Jersey’s Attorney General, and the President of the Commonwealth Association of Legislative Counsel, both brought up our work at a Commonwealth Law Ministers’ Meeting. We have been invited to speak at various legal and technology conferences across the world (remotely in many cases), and we have worked with teams from as far away as Singapore, Australia and Canada (and elsewhere).

That has brought direct benefits of collaboration. By achieving this level of international recognition, the project has also contributed to enhancing Jersey’s reputation as a world-leading innovator in both the legal and digital fields.

1.3 The rest of this report

The rest of this report covers the main results of the project so far, giving more detail but also links that will “show” rather than “tell” –

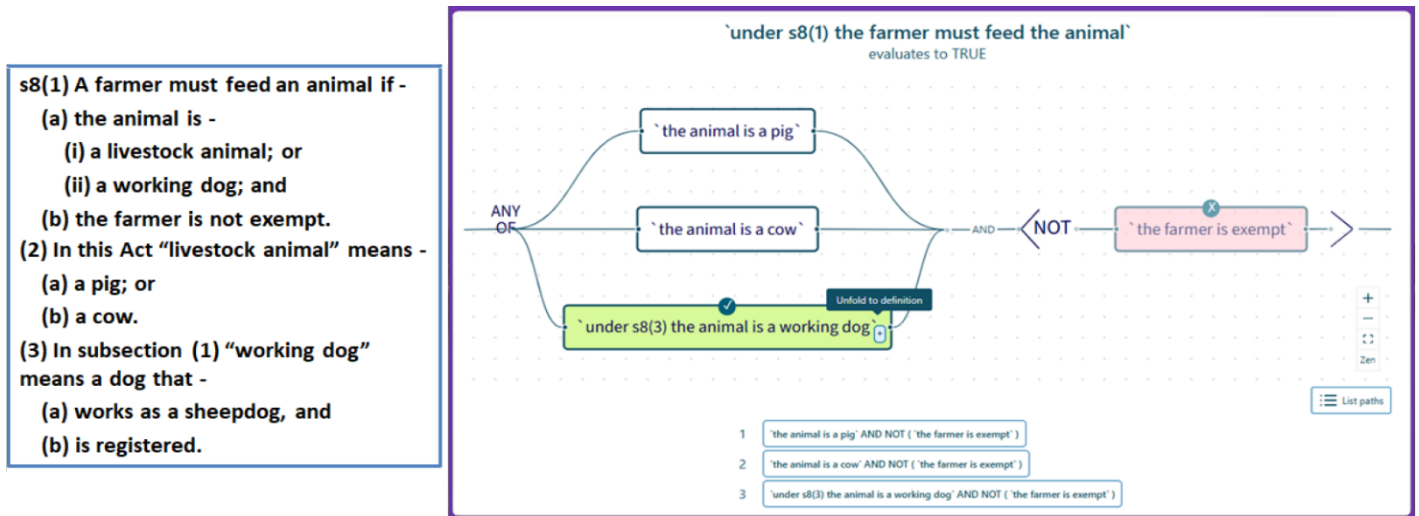
- the L4 logic [visualiser](#),
- our [definitions](#) tool,
- our work on a future [drafting](#) tool,
- the ways we have used [AI](#),
- our drafting [guidance](#) and training,
- our illustrative tool for drafting [instructions](#), and
- the way we have [published](#) our output.

For readers who want to know more about the background to this work, the report then adds some [background](#) detail on –

- legislative [drafting](#) and the Legislative Drafting Office,
- the “[Rules as Code](#)” movement,
- our approach to helping readers rather than [automating decision-making](#),
- our preference for [logic programming](#) approaches, and
- our focus on [new](#) legislation rather than the whole of the existing statute book.

Then there is a list of [key concepts and agencies](#) relevant to this report. The report [concludes](#) by summarising the achievements so far and looking forward to future work of the project.

2 Logic visualiser – interactive map of logical structure of a legislative provision



What if legislative drafters could produce an interactive map of the conditions for the legal effect of a provision they are drafting?

Our focus has been to ensure that what we produce can be used by legislative drafters without having to download any software, pay any fee or learn to code. We are particularly keen on illustrative, interactive demonstrations that help legislative drafters (and developers) to understand the sorts of tools they could have and the sorts of benefits those could bring.

2.1 Using L4 in an interactive logic visualiser of conditions for an effect

This is a free, web-based tool that legislative drafters can use to generate an interactive map of the logical structure of their draft. It is hosted by [Legalese](#) and we would encourage readers to try it out for themselves, starting with [this example](#). To see how to use it, watch this [demonstration video](#), or look at [our newsletter article](#), or [our slides](#), or these [instructions](#).

- The drafter makes an if-then version of the draft, in English text drawn from the text of the draft and paragraphed and indented like the draft, to pick out the legal effect and conditions for its application.
- The drafter then automatically turns that into basic L4 code. Because the Legislative Drafting Office uses Word, we have a Word macro to do this job, but any other automated system could be used, and at some point the L4 website might incorporate this function.
- The drafter copies that L4 version into the visualiser, which then automatically generates the interactive map of the conditions and effect.
- The drafter, or another user like the policy officer, can then click on boxes containing the conditions to indicate whether they are met or not (they turn green or red accordingly, or white if there is no answer).

- The visualiser then automatically updates the map to show whether the legal effect applies or not.
- The user can give different answers to check that various fact scenarios all have the desired results when the draft provision is applied to them.

This is a freely accessible tool that does not require any downloading, to encourage drafters to try it out, so that they can understand what we mean when we talk about enabling a computer to read the logical structure of their drafts. The flipside of that is that it cannot be used for actual live drafts because it is a public site and those drafts are confidential. But the point is that L4 is open source and drafting offices should be able to produce their own confidential versions of the visualiser if they decide to go down that road (or find developers to produce a local equivalent, or whatever).

We have also looked at expanding on the use of the if-then rewritings. Jersey's government has various computer systems that implement policies that include legislation. We are required to keep the details confidential, but we are talking to people in government working on systems that use rules engines to handle models of the logic of the rules in the relevant legislation. That includes creating pseudocode representations of that logic which are similar to our if-then versions. We are interested in whether the rules can be visualised using the pseudocode (learning from our L4 visualiser) and whether policy officers and legislative drafters can then contribute to updating the models as part of developing future amendments to the legislation.

For those interested in the technical background – [L4](#) is a “domain specific language”, or DSL, for law. With additional funding from the [Jersey Legal Information Board](#), we were able to work with a global team of experts at the [Centre for Digital Law](#) at Singapore Management University on several ideas to see what we could do with L4 for our project. SMU have now finished their research, and L4 is being taken forward by [Legalese](#) whose L4 team can now be found on [Discord](#).

2.2 Using L4 more fully

The description above is about deliberately under-using L4, so that legislative drafters can achieve something worthwhile just with the if-then analysis that fits how they already draft (without learning to code). That means we are just getting drafters to use the equivalent of “propositional” logic in the L4. But L4 itself is a full computer language, capable of doing much more than if-then. We have worked with the L4 team to ensure they understand how legislation works.

ielegante / l4-jerseyCharities Public

Couldn't load subscription status. [Retry](#) Fork 2

<> Code Issues Pull requests 1 Actions Projects Security Insights

main 1 Branch 0 Tags Go to file Code

ielegante [feat] improvements + fixed oversights: 9cc94fc · 3 months ago 6 Commits

README MIT license

l4-jerseyCharities

A WIP codification of the Jersey Charities Law, written in L4 to demonstrate the capabilities of the L4 Domain Specific Language.

The `reference` folder contains the primary and subsidiary legislation that this was based of.

The `thinking` folder contains docs that document the thinking behind the codification approach.

The `docs` folder contains guides and other docs that were created in the process of codification, in a bid to create guides that are useful to both humans, and also large language models. The latter follows the lms.txt format.

This codification represents a first attempt at codifying the Jersey Charities Law, from the

About

No description, website, or topics

Readme MIT license Activity 2 stars 0 watching 2 forks Report repository

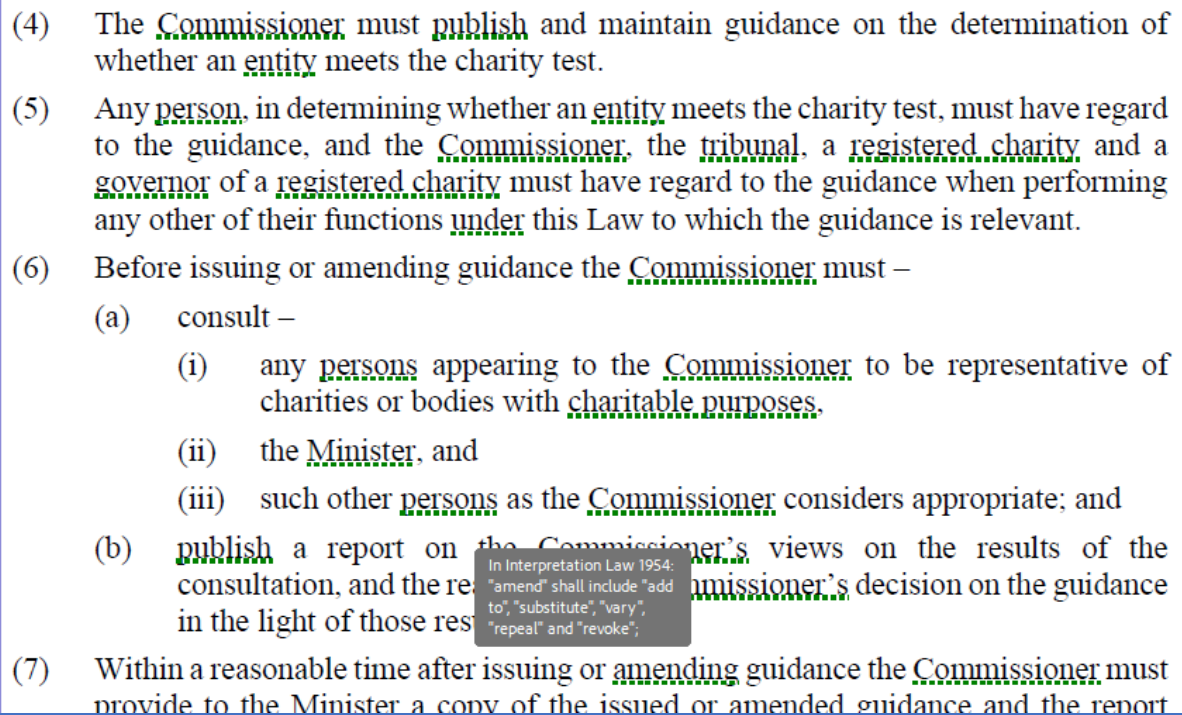
Releases

No releases published

In turn the L4 team have produced [a first go at an L4 version](#) of the whole of the [Charities \(Jersey\) Law 2014](#). We chose the charities legislation because it is a fairly self-contained set with the Law, some Regulations and some Orders, which mesh together without depending too much on other legislation. We have also been encouraged by interest from the [Jersey Charity Commissioner's office](#), who had to produce their own model of the legislation when they first needed to create their website and supporting IT systems before the charities register could open in 2018.

We are planning to work with the Commissioner's office in future to see how they could make use of a full L4 encoding of their legislation to benefit the charity sector in Jersey. With Google now promoting AI-generated answers to searches, it is increasingly likely that people who want to know whether they can register a charity (or anything else about regulation of charities) will first see an AI-generated answer that may contain hallucinations but look convincing. That is why we are keen to try to use a logic-based approach like L4's to improve the reliability of the answers that the public finds. That obviously also applies to many other areas where the public (and businesses) search for information about the regulation of their daily lives, and we plan to expand field by field beyond charities if we have some success there.

3 Tools to mark up definitions and help drafting generally

- 
- (4) The Commissioner must publish and maintain guidance on the determination of whether an entity meets the charity test.
- (5) Any person, in determining whether an entity meets the charity test, must have regard to the guidance, and the Commissioner, the tribunal, a registered charity and a governor of a registered charity must have regard to the guidance when performing any other of their functions under this Law to which the guidance is relevant.
- (6) Before issuing or amending guidance the Commissioner must –
- (a) consult –
 - (i) any persons appearing to the Commissioner to be representative of charities or bodies with charitable purposes,
 - (ii) the Minister, and
 - (iii) such other persons as the Commissioner considers appropriate; and
 - (b) publish a report on the Commissioner's views on the results of the consultation, and the Commissioner's decision on the guidance in the light of those results.
- (7) Within a reasonable time after issuing or amending guidance the Commissioner must provide to the Minister a copy of the issued or amended guidance and the report
- In Interpretation Law 1954: "amend" shall include "add to", "substitute", "vary", "repeal" and "revoke";

3.1 Pop-ups and links for definitions – a tool in the testing stage

3.1.1 The definitions tool

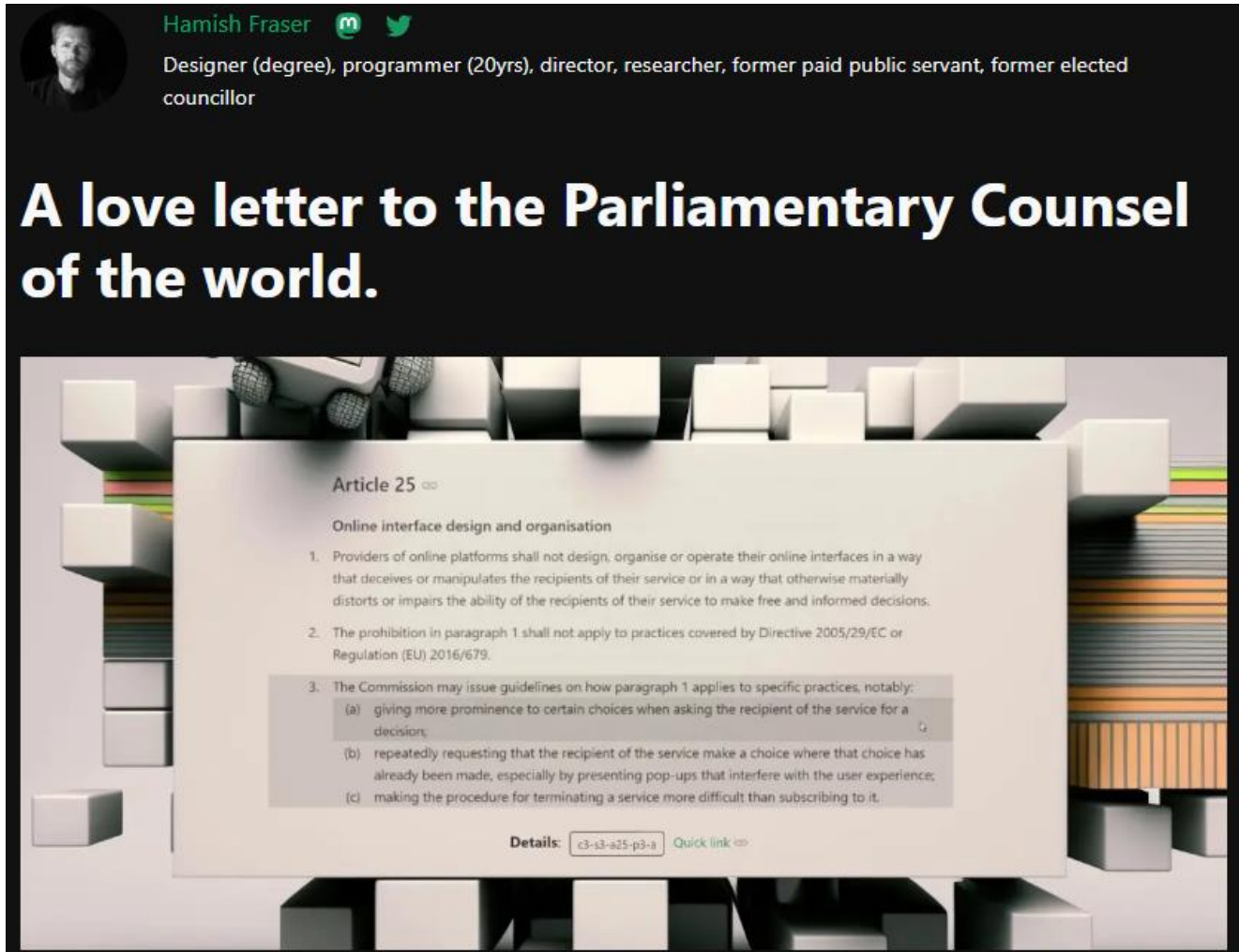
The Computer-Readable Legislation Project has worked with the Legislative Drafting Office to develop a tool (illustrated above) that some of the Office's drafters are currently testing, to mark up each defined term with a pop-up that shows the definition and gives a link back to wherever the definition appears in the draft (see this [video](#)).

- Importantly, the tool also covers definitions in the [Interpretation \(Jersey\) Law 1954](#), which apply in every other Law unless they are positively displaced (as in other Commonwealth countries).
- It also covers definitions in a Law which then flow through into Regulations or an Order made under that Law. In Jersey this is under Article 10 of our Interpretation Law, but the Interpretation Acts of most other Commonwealth countries have similar provisions.
- Many readers of a piece of legislation would otherwise be left completely unaware of either of these additional sources of definitions.

We have had positive feedback from policy officers where drafters have applied the tool to active drafts. We are now testing the system further to see whether it can be extended to legislative drafts in Assembly propositions and eventually (we hope, but cannot promise) to new legislation published on the jerseylaw.je website.

3.1.2 Paragraph IDs

The system is built on new identifiers that are automatically applied to each paragraph of text. The identifiers reflect the structure and are readable by both humans and computers. They are likely to enable us to develop further tools and features in future.



This came from contacts the project made early on with Hamish Fraser, in New Zealand, who works with [Syncopate](#). In February 2023, Hamish sent a [Love letter to Parliamentary Counsel of the World](#), and some [sequels](#) including “Happy Valentine’s Day, Nerds”. The idea is that every paragraph of text should have its own unique ID which captures the structure (of sections and sub-sections, etc.) in a human-readable way. Once you do that, unexpected possibilities open up. Hamish and colleagues used their idea to make [DocRef](#) with nested, understandable and manageable IDs that produce various benefits. The Computer-Readable Legislation Project plans to continue working with our New Zealand friends to see what other uses of these IDs could promote our aims.

3.1.3 Manual markup during drafting, not later automated markup

Several of the Legal Information Institutes in other countries (the equivalent of the Jersey Legal Information Board, but publishing unofficial versions of legislation) have

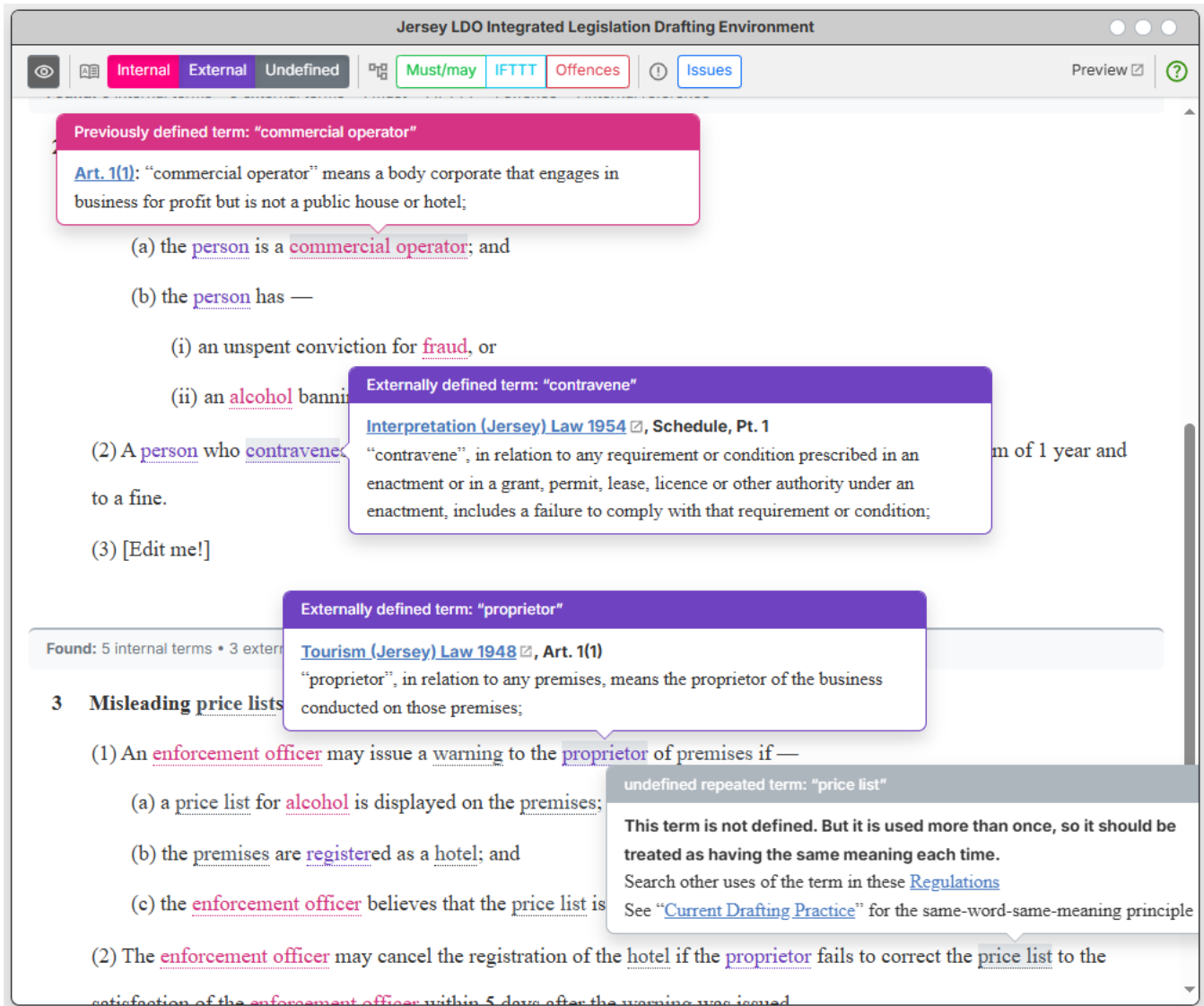
used automated tools to go back over a country’s entire existing statute book and apply pop-ups and links from defined terms to their corresponding definitions. This is clearly of some benefit in an unofficial version of the legislation, if it comes with a large health warning.

- The main problem is that the automated system will pick up many false positives – words in the text that match the defined term but that a human reader would realise are not really instances of that term. If this is being done across the whole statute book it is not going to be practical to have a human expert then check every marked-up word to see if the markup should be removed.
- Similarly, there may be a problem of false negatives in that definitions are usually meant to cover grammatical variations of the defined term, such as “registration” and “register” (Jersey has that rule in Article 2(c) of the [Interpretation \(Jersey\) Law 1954](#)). That means the automated system has to use a highly sophisticated search which catches the additional forms of the word without worsening the false positive problem.
- Even if technical solutions can be found for both those issues, there is still an underlying legal problem that cannot be fixed automatically. That is that in older legislation, still on the statute books, there will be mistakes where the defined term has been inadvertently used in a different sense. That reflects the fact that, in the days of handwriting on paper before Word search and systematic checking, drafters could not be sure they had used defined terms correctly even in a fresh draft. That was why definition provisions used to start with “In this Law, *unless the context otherwise requires*”, because that also served as an escape route for mistakes. In recent years Jersey’s Legislative Drafting Office has stopped using that wording because it is now reasonable to expect the drafter, helped by the checkers, to make sure defined terms are used properly in a new draft. The problem is that in the older legislation only a court could decide whether a questioned expression does attract the defined meaning – an automated system certainly cannot do that, and even a modern expert cannot claim certainty (usurping the court’s function).

It is, of course, more work to mark up each instance of a defined term manually, and the benefit is more restricted if it cannot be done for the whole existing statute book.

- But the testing being carried out by the Legislative Drafting Office on the tool is to see whether this manual markup is practical when done as part of the drafting process. So far the results seem positive, in that the effort is repaid by keeping the drafter alert to the definitions as they go along and helping reviewers to spot errors. The testing also identifies ways in which the tool might be refined to make the manual process as easy as possible.
- If the tool can sensibly be rolled out into full use, so that the links appear in the published legislation, it will be on an incremental basis, in that the links will only appear in new legislation (or older legislation that is re-enacted for the purpose). But that is very similar to the Computer-Readable Legislation Project’s approach to “Rules as Code”, as explained in the [background detail](#) section below (which also points out how Jersey might be a good fit for the incremental approach).

3.2 Groundwork for a drafting tool



Going further than the definitions tool, we have also produced an [interactive mock-up](#) of a more wide-ranging use of pop-ups and links, which we [explained in our newsletter](#). As illustrated above, this shows how it could be useful for a pop-up tool to cover not just defined terms but also repeated undefined terms.

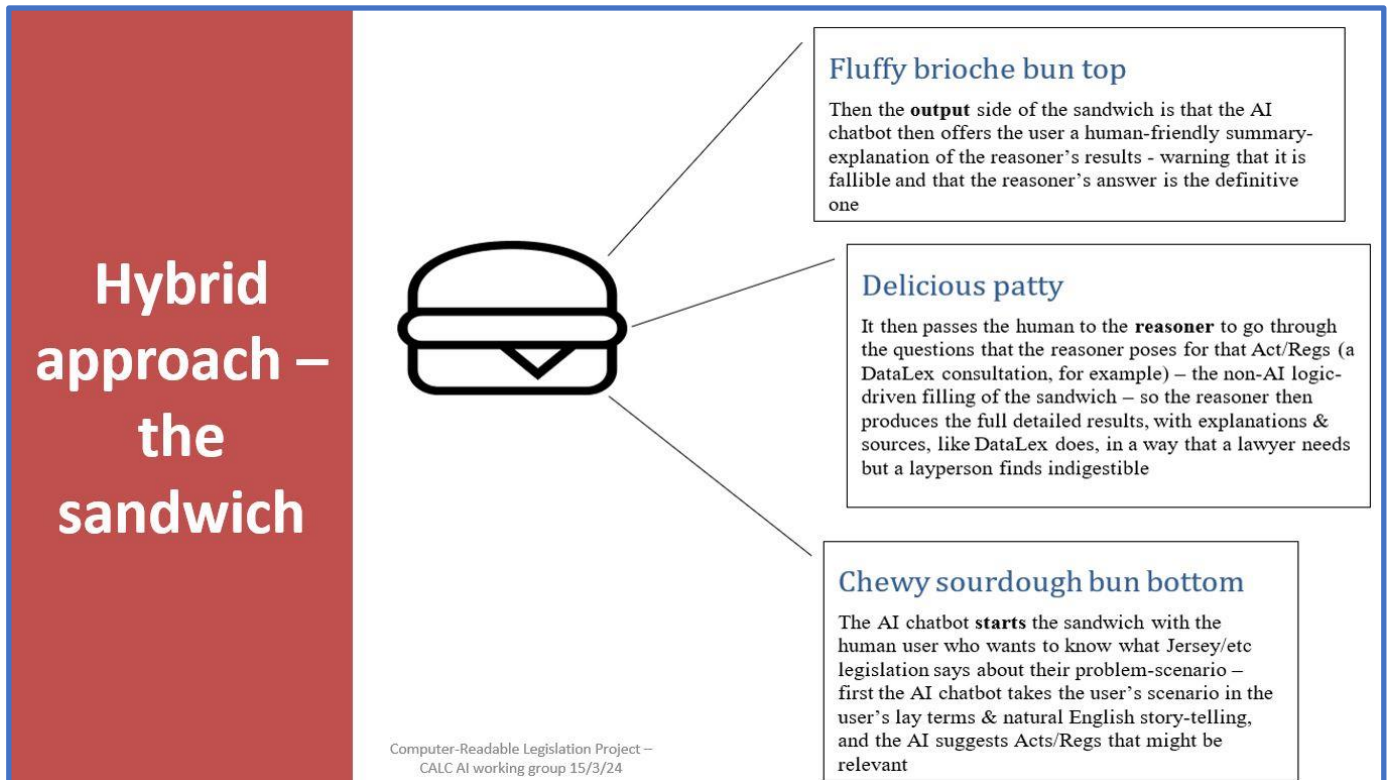
The point there is that, while they are drafting a Law, the legislative drafter is relying on the interpretation principle that the same word should be taken to have the same meaning, while a different word implies a different meaning (even where they might otherwise just be assumed to be synonyms). This creates a very disciplined style of writing, which means readers should be looking for patterns of repeated words even when those words are not defined.

We set out above how a Law will not refer to the [Interpretation \(Jersey\) Law 1954](#), which leaves non-expert readers unaware that its definitions apply to the particular Law in front of them (so the definition tool is designed to cure the effect of unhelpfully hiding the Interpretation Law). There is a similar problem in that this same-word principle is

not explained on the face of each Law, and is not even mentioned in the Interpretation Law (nor in those of most other Commonwealth countries), so non-expert readers will not know about it. The mock-up illustrates what would happen if drafters marked up the instances where they were repeating an undefined expression. For the way in which it can be important for an undefined term to have the same meaning each time, see our [explanation using an example of “fish farmer”](#).

We have also published additional ideas about how legislative drafters could have writing tools that match the usefulness to coders of their “integrated development environments” or IDEs – perhaps an integrated legislative drafting environment or ILDE. For more on this, see the relevant [page in our website](#) and [documents in our repository](#).

4 AI – artificial intelligence



4.1 Introduction

ChatGPT brought one form of Artificial Intelligence to the public's attention just after we settled the terms of our Computer-Readable Legislation Project. Fortunately our plans were flexible enough, and focused on the outcomes rather than the specific technology, so that we were able to adapt the project to take advantage of what AI offers.

We have found ways to use generative AI (particularly chatbots using Large Language Models, or "LLMs") in ways that play to its strengths, but avoid relying on it for jobs that it is not appropriate for. While LLMs continue to generate the most probable text, "hallucinations" will remain a feature, making them unreliable for use on their own to work out how legislation applies to given facts. Our work on AI includes the following.

- We [published](#) our take on how chatbots can be used in a hybrid (or neuro-symbolic) "[AI sandwich](#)" as an interface for a human to use a reasoning system, and then worked with the L4 team to test a version of that.
- We helped set up the AI working group of the Commonwealth Association of Legislative Counsel, and have liaised with the Jersey AI Forum and the government's Digital Services group on AI.

A chatbot that uses generative AI for the reasoning will inherently remain not fully reliable. But it can be useful if it comes with adequate health warnings and the user is likely to check the results. We have worked to make Jersey's legislation available to

everyone so that local and other developers should be able to do produce chatbots for our legislation –

- we have spoken to local firms about their chatbots,
- we have put out guidance to help them see how our legislation works, and
- we drove the move to issue a workable copyright licence for our legislation so that developers can use it.

There are already examples of AI-powered chatbots developed by local firms, such as [AskBailey](#), [Planning.je](#) and [AskMax](#), which use Jersey legislation with varying degrees of reliability and transparency. We find they can struggle with key requirements such as being able to apply definitions. As is to be expected with LLMs, when they do seem to have applied a definition correctly it is not possible to check whether or how they actually did so.

It is useful to compare those chatbots with [Reggie](#), the chatbot being developed (still in beta) by the Jersey Financial Services Commission, which comes with a lengthy [user guide](#). JFSC started with plans to produce computer-readable guidance, but they were persuaded by Digital Jersey to switch from that into working on their own AI chatbot. JFSC is in a position to have its own purely AI chatbot because the primary users will be its regulated firms, so JFSC can tell the firms not to rely on the chatbot and can discipline them if they do so. By contrast, the Legislative Drafting Office is drafting legislation that covers all fields and is published through the Jersey Legal Information Board for all members of the public. Neither our Office nor JLIB have any disciplinary authority over the public to tell them how to use a chatbot. So if our Office were to endorse a chatbot it would need to be reliable, by using the LLM just as an interface for the users to deal with a reliable logic-driven system underneath. So our project has –

- published explanations about our proposed solution for chatbots, and ensured developers can access the legislation,
- worked with Digital Jersey and Google Cloud AI on using AI to analyse legislation,
- worked with the L4 team on using AI to supplement the logical reasoning of L4.

This work is described in more detail in the following sections.

By the way, we did not use AI to write this report, but we did use it to generate 3 of the graphics and to check the summary.

4.2 Enabling AI systems to ingest our statute book

We have encouraged other people to apply machine learning to our legislation, and other forms of AI beyond LLMs.

4.2.1 Licence and API

AI chatbot developers, and machine learning researchers, need to be able to have their systems ingest the whole of our current statute book, and to be able to do so legally and without crashing our legislation site. After we were approached by firms in Jersey and outside, we took the initiative, working with the Law Officers' Department, the

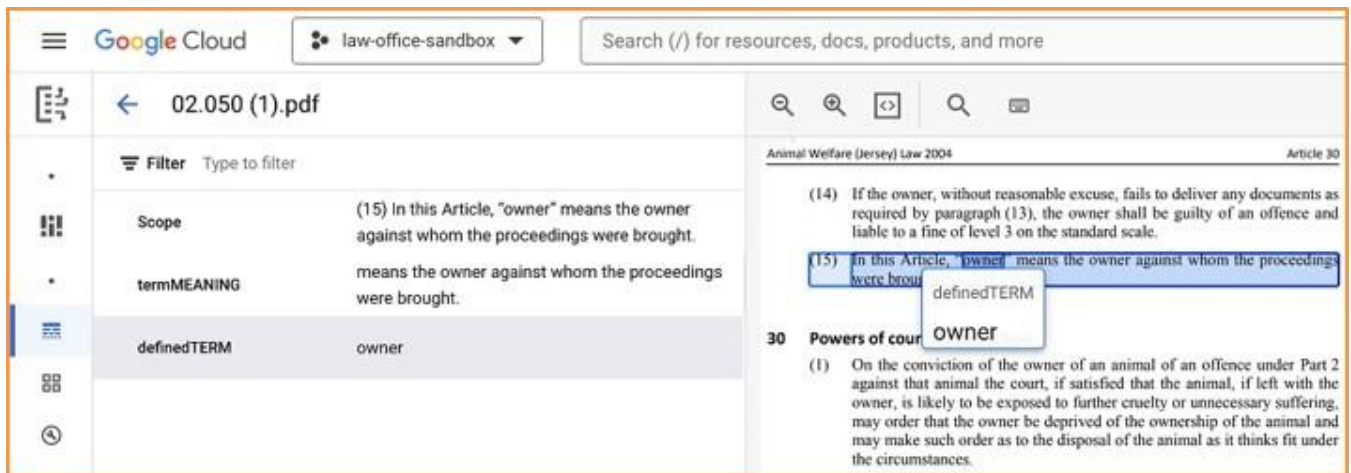
States Greffe and the Jersey Legal Information Board, to produce an [open copyright licence](#) for our legislation on [jerseylaw.je](#), along with an [explanation](#) of how Jersey’s copyright law applies to our legislation. JLIB set up an access point (an “application programming interface”), to avoid problems caused by AI developers scraping the site (details of the API are available to organisations on request from JLIB).

4.2.2 Notes for developers of chatbots for Jersey legislation



We produced [notes for developers of chatbots](#) to help them see how Jersey legislation works (much of which also applies to legislation in other parts of the common law world) and where LLMs can fall down. The notes go into how to find the different types of legislation on [jerseylaw.je](#), what the different versions are for, and the copyright licence and the API for bulk access to the legislation. The notes also point out that a chatbot needs to be clear about whether the user is asking about the law as it is today (“can I build a shed?”) or as it was when some past event happened (“do I have to knock down the shed I built last year?”) – and explains that the website’s default is to present today’s legislation, but that it can also do so for any past date back to 2019 (but in a way designed for human readers rather than AIs).

4.3 Google Cloud AI with Digital Jersey



We worked with [Digital Jersey](#) on a project to use [Google Cloud's Document AI](#) to identify elements in legislation such as offences. That taught us useful lessons on techniques in training AI. But it also identified past poor drafting practices on offences which gave an opportunity for improvement. We looked into ways to fix problems in older drafts using editorial powers, and we organised a meeting to discuss better practice in future, followed by a Saver page (the Legislative Drafting Office's knowledge management wiki) to embed the better practice.

On AI capabilities, the main lesson was that more training (as opposed to providing structured input in the form of computer-readable legislation) does not necessarily give better results. The base model gave a reasonable start but struggled with consistency. We then trained the model on ten human-marked examples, expecting this to make better results. Instead, the opposite happened - its performance *declined*.

- This experience showed that machine learning doesn't follow a straight, human-like path of improvement. We often assume *more training means better results*, but AI models learn through patterns, not reasoning. When new data are added, they can shift how the model interprets earlier patterns. This means improvement is non-linear: progress can jump, stall or reverse depending on how the data and model interact.
- In our case, it might be that the model recalibrated too heavily around the small set of training examples and started losing the balance it had before. That doesn't mean the product failed or that the data were bad. It means training an AI model is a sensitive process: even good data can cause a model to over-adjust or "forget" what it previously did well. This phenomenon is known in machine learning as *catastrophic forgetting*, but in simple terms, it means the model concentrated too much on the new examples at the expense of everything else.

4.4 Making an AI sandwich with L4 as the reasoner filling

The screenshot shows a web browser at `https://legalese/jl4_ai.html`. The interface has a dark theme. At the top, there's a "History" button and a "Settings" gear icon. A blue input box contains the text: "I was born in Gibraltar in 2000 to a British father and a Spanish mother, when they were both settled in Cayman". Below this, a section titled "Assessing policies ..." shows "1. Relevant Policy: The British Citizen Act". It then says "Applying prompt and context information:" followed by several fields: `p_name: Person`, `p_birthPlace: Gibraltar`, `p_birthDate: 2000-01-01`, `f_settledIn: Cayman Islands`, `f_knownIn: Cayman Islands`, `m_settledIn: Cayman Islands`, and `m_knownBritish: false`. A "Decision" box shows `result: FALSE`. To the right, a dropdown menu for "Search model name" is open, showing options: `claude-opus-4-20250`, `claude-sonnet-4-2025`, `✓ claude-3-7-sonnet-20`, and `claude-3-5-sonnet-20`. Below the decision, a message says: "I'll help you understand your citizenship status based on your birth circumstances. Let me evaluate this for you. Based on the policy evaluation of your circumstances: The person born in Gibraltar in 2000 to a British father and a Spanish mother who were settled in the Cayman Islands is NOT a British citizen. The reasoning behind this determination: 1. Gibraltar is confirmed as a British Overseas Territory and qualifies as a 'qualifying territory' for citizenship purposes. 2. The person's birth circumstances match the requirements for British nationality legislation. 3. Two main citizenship pathways were evaluated: My International Upbringing in Gibraltar and Cayman". At the bottom, there's a "Need more details?" button and a "New" button.

This was an experiment carried out with the L4 team who created a demonstration website where a user can ask a question about the application of the British citizenship rules just by entering the facts of their (imaginary) scenario, worded however comes naturally to them (and the user can pick different AI models). The demonstration site uses paid AI services, so it is not open to the public, but anyone interested in trying it out for free can contact Legalese (through Discord, from their [website](#)). The idea of the demonstration was to test the “[AI sandwich](#)” concept described above, in which the AI chatbot is the interface between a logical reasoner and a human user’s natural English.

- The results were that the AI is very good at taking the human user’s natural language **input**, finding the correct rules in the L4 set, extracting the relevant facts from the narrative description, and sparking the L4. It is particularly helpful in that, if L4 is told the relevant territory’s name is “Cayman Islands” then a purely L4 system will not recognise just “Cayman” as an input. But, as shown in the example in the illustration above, an AI can take “Cayman” as an input and match it correctly to the L4’s requirement to see “Cayman Islands”.
- As expected, the L4 **reasoner** then worked reliably as long as the input was correct, because it is the logic-based middle layer of the AI sandwich, so there is no chance of hallucination. For this experiment, the only provision coded into L4 was s.1(1) of

the British Nationality Act 1981, whereas of course a real app would need all the rest of the relevant legislation coded too.

- Most of the problems came in the third layer of the AI sandwich, when the AI takes the L4's **output** and explains it in more natural English to the human user. At this point it became extremely difficult to get any of the standard AI models to restrain themselves from going beyond translation of the L4 answer into trying to answer the question themselves. At that point they would frequently hallucinate, drawing in their misunderstandings about other nationality law provisions, but in the usual convincing way. The non-expert user would not be able to see any sign of where the L4-based answer ended and the AI-invented answer started. Equally it seemed extremely difficult to let the AI add its own extra material but then clearly flag which part of the answer was which.

4.5 Local and global

As with our other work, the project has engaged both locally and globally with AI.

- We joined the [Jersey AI Forum](#) at its foundation and gave a presentation to their Legal and regulatory sub-group.
- We have also encouraged others to apply machine learning to our legislation, and other forms of AI beyond LLMs, including the [Regulatory Genome Project](#) at Cambridge University and the associated company [RegGenome](#).
- At the same time we have engaged with some of the Jersey tech firms working on legislation chatbots (see above).
- We also contributed to setting up the Commonwealth Association of Legislative Counsel's working group on AI, and gave one of the first talks to share our experience with AI with other drafting offices.
- Two of our members co-wrote a [journal paper](#) with the founder of the CALC AI working group, setting out our "AI sandwich" model, so far cited in 8 other research publications.
- The Computer-Readable Legislation Project has also more broadly spurred on the Legislative Drafting Office to keep abreast of the implications of AI for its wider work. Two staff from the Legislative Drafting Office attended the European University Institute's 2023 week-long AI & Law summer school. The Office has held internal sessions on using AI, and it has engaged in the government-wide groups looking at use of AI.

5 Non-technological drafting guidance



5.1 Guidance and training

In the course of the project we have gained insights from our work on logic and computing. That has led us to produce [guidance and training](#) to show how legislative drafters can further improve our drafting of the English text of legislation, regardless of what technology we use.

- For Jersey’s Legislative Drafting Office, we have run multiple training sessions with the drafters and other members of the team, and we have produced guidance tailored to Jersey and created pages on Saver (the office’s internal knowledge management wiki). We have incorporated this, and the technological lessons of the project, into the training for the Legislative Drafting Office’s trainee drafters, making them the world’s first “born digital” legislative drafters.
- What we have learnt is adaptable to other Commonwealth drafting offices. So we have also run training and demonstration sessions at conferences of the Commonwealth Association of Legislative Counsel, and written articles for their Loophole journal and items for their Newsletter.

5.2 Types of non-technological drafting improvements

These drafting improvements do not depend on any particular technology being developed. Instead they are about things like –

- always ensuring it is clear what should happen if conditions are not met, or if something that “must” (or “must not”) be done is done (or not done), or if a power is exercised correctly or incorrectly,
- ensuring the tenses we use for verbs are clear enough to signal the difference between what has to have happened before an effect bites and what has to be happening while the effect bites,
- avoiding syntactic ambiguity while still explaining to policy officers that some degree of semantic vagueness is unavoidable and has to be balanced against the risks of over-specification generating its own unexpected consequences, and
- being clear about the type of function being created in each provision and their components (and the structures built from their relationships to each other), such as –
 - creating a criminal offence,
 - imposing an obligation or prohibition on a person,
 - giving a person a permission, as an exception from a prohibition,
 - giving a person a power to change the legal status of another person,
 - defining a term (exhaustively, inclusively or exclusively),
 - setting a condition for something to count as a valid exercise of a power or a valid fulfilment of an obligation,
 - granting a power to make subsidiary legislation,
 - commencing, repealing or amending legislative provisions, or
 - other constitutive provisions.

5.3 An Interpretation Law for the digital age

This work has also led us to [recommend changes](#) that could be introduced to turn the [Interpretation \(Jersey\) Law 1954](#) into the first interpretation legislation in the common law world that would be fit for the digital age, when it is overhauled as part of the Legislative Drafting Office’s ongoing work.

- In particular, we have found issues with the definition of “person”, which is the most fundamental unit in the logical structures of our legislation. In Jersey, as in almost all other Commonwealth jurisdictions, the definition of “person” still remains largely unchanged from a [UK model from 1889](#). The definition unhelpfully includes an “unincorporated” body of persons, which seems meaningless but looks as if it is somehow intended to cover groups that do not have separate legal personality. In practice, drafters always use “person” to mean either a human or something else that has separate legal personality.
- We have also looked at the use of “function”, “power”, “contravene”, “must” and “may”, and the “always speaking” principle, to see whether the new Interpretation Law could be updated to clarify modern uses and make it easier for computers as well as humans to read the legislation.
- We are hoping this work might be taken forward next year once the current pre-election drafting is completed.

6 Docassemble demo with “Common Legislative Solutions”

6 Licensing schemes	COMMON LEGISLATIVE SOLUTIONS: a guide to tackling recurring policy issues in legislation
This legislative solution involves making the doing of an otherwise lawful activity unlawful, unless done in pursuance of (and in accordance with) a licence. It is, therefore, a tool used to regulate and monitor a particular activity – or in other words to decide how it should be done.	
Description of legislative solution	are met (e.g. registration of doctors by the General Medical Council). In such cases, the difference between “licensing” and “registration” may be minimal.
The power to grant licences may be conferred on a part of government or on another body or person.	Elements of the legislative solution
Licensing regimes range from the very simple (e.g. a dog licence or a TV licence) to the complex (e.g. planning permission or the licensing of the supply of electricity).	1. Activity to be regulated
Related legislative solutions	1.1 What is the activity that is to be subject to the licensing regime? NB: there is an important link here to the issue of enforcement considered below – doing the activity without a licence is almost invariably criminalised, so the activity must be defined with sufficient clarity to form the basis of an offence.
Notification regime	1.2 Should there be any exceptions, i.e. any persons who may do the activity without a licence or any other cases in which the activity
A close relation of this legislative solution is a notification regime, i.e. a regime that requires a person to notify a regulator of the person's intention to do a particular kind of activity, with the regulator having certain powers in relation to the doing of the activity. An example of this	

Something that many policy officers find difficult is writing “drafting instructions” to set out what legal changes they need the legislative drafters to create. The Legislative Drafting Office offers training, but technology could help more. Using [Docassemble](#), we have produced a free, online, interactive, illustrative example of an app (see [below](#)) called “[Licensing - Common Legislative Solutions - Jersey version](#)”. The app guides a policy officer, who wants new legislation to set up a licensing scheme, through standard questions about the policy. The policy officer’s answers would take them to further questions, ending with the app giving them the appropriate model legislative provision. The Computer-Readable Legislation Project’s contribution is that the model provision would be accompanied by an interactive visualisation (see [Logic visualiser](#) above). The policy officer would use that to check that the suggested model provision would do what was wanted.

6.1 Background

Our project supported the Legislative Drafting Office in setting up a “coding club” for staff. The club decided to learn [Docassemble](#), because two team members had previously used it to develop a successful [app currently in use](#) for instructing officers to write drafting instructions for annual fee increases (explained in this [video](#)). Docassemble is a free, web-based, open-source expert system for guided interviews from which documents can be assembled.

Separately, the Legislative Drafting Office had also been working on a Jersey adaptation of “Common Legislative Solutions” from the drafting offices of [UK](#), [Scotland](#) and [Wales](#). Common Legislative Solutions is a set of questions and guidance that takes policymakers through a set of questions about typical issues that come up in many legislative drafting proposals, such as setting up statutory bodies, search powers, appeals, licensing schemes, criminal offences, and so on. Common Legislative

Solutions was itself inspired by the concept of “pattern languages” from computing (see Norbury, [Loophole, Sept 2018](#)), and two of our assistants had previously written published papers on patterns in law (see [Qiu](#) and [Diver](#)).

The Legislative Drafting Office’s idea is to combine two other sources with Common Legislative Solutions to produce useful guidance for Jersey civil servants (and for the States Greffe staff who support backbench Assembly members when they want to develop legislative propositions). The two other sources are –

- the Legislative Drafting Office’s training notes for Jersey civil servants on how to write drafting instructions, and
- “[Making Laws That Work](#): How Laws Fail and How We Can Do Better” (by David Goddard, a judge in New Zealand’s Court of Appeal), particularly the [checklists](#) for assessing whether legislating is going to be the best approach.

The end point in each subject in Common Legislative Solutions is a set of examples of appropriate provisions drawn from existing legislation. In Jersey we are thinking of supplementing that with suggested model provisions in line with [Jersey’s Current Drafting Practice](#). The Computer-Readable Legislation Project will in turn supplement those with if-then versions of the model provisions, which can be fed into our interactive [Logic visualiser](#).

The coding club used [Docassemble](#) to learn how to convert early drafts of parts of this Jersey version of Common Legislative Solutions from a static guidance document into an app that civil servants could use. The app would help them formulate their policy, write their drafting instructions, and get an idea of the sorts of provision that the Legislative Drafting Office could draft for them, in a form that they could use to check that the provision produces the desired results for different fact scenarios.

6.2 The mock-up of an app with if-then version & logic map

To illustrate this idea, we have produced an interactive mock-up “[Licensing - Common Legislative Solutions - Jersey version](#)”, which we invite readers to try out.

This is a cut down version of what would be in the final app, with only certain routes working, to demonstrate reaching the point where the app provides something that can be used in the logic visualiser.

- So the mock-up of the app just imagines a policymaker wanting a licensing scheme, in which conditions can be imposed on licences (the mock-up ends if the user chooses not to have conditions).
- It asks the user who will grant the licences, and then focuses on the important, but often overlooked, question of what the policymaker wants the consequence to be when a licence-holder breaches a condition on the licence.
- The options offered are that the breach could be an offence, or it could trigger a power or duty for the licence-granter to revoke the licence, or the breach itself could result in the expiry of the licence by operation of law.
- The full app would contain the explanations of each of these from the eventual full Jersey version of Common Legislative Solutions.

Licensing - Common Legislative Solutions - Jersey version </>

Consequences

What are the consequences for breaching the conditions?

☐ Offence

☒ Expiry of licence by operation of law

☐ Licence-granter must revoke licence

☐ Licence-granter may revoke licence

Continue

Legislative Drafting Office, Jersey

< Back Licensing - Common Legislative Solutions - Jersey version </>

Result

Please look at this model draft provision for the answer: **Expiry of licence by operation of law**, with its if-then rendering of the conditions and effect for the model provision. Also follow the link to the interactive visualiser - at the L4 site press visualise then click on the boxes to try out how different answers about the conditions will change the effect.

Are you happy you understand the model provision and that the visualiser gives the answers you want? Please download all relevant documents BEFORE you answer this question.

Yes No

The following document has been created for you.

Click below to download the draft provision in Word.

Download

[DOCX](#) (for editing; requires Microsoft Word or compatible application)

Legislative Drafting Office, Jersey

The mock-up is set up so that the user picks one of these and is then presented with a document that contains –

- the appropriate model legislative provision,
- its corresponding if-then version,
- the L4 code to represent that version, and
- instructions to feed that L4 code into the visualiser – where, for the sake of the example, the link goes to the code already in the visualiser.

The idea is to offer an interactive experience (as “show” is better than “tell”) that gives an idea of how a fully worked up Docassemble app could transform Common Legislative Solutions into something much more ambitious than a static guidance document.

7 Making our work freely accessible with global impact

7.1 Working in the open

As well as ensuring the benefit of our work is felt by the legislative drafters and editorial & technology team in Jersey’s Legislative Drafting Office, we have shared our work internationally and attempted to [work in the open](#).

- We have done this in several ways, including making all our work freely accessible in a [repository on the Open Science Foundation](#).
- We have also continually disseminated our findings as we have gone along, through –
 - our [website on GitHub](#) (set up by one of the project’s assistants and maintained by Legislative Drafting Office staff and project volunteers, all at no cost),
 - our Substack newsletter “[Digital Legislation](#)”,
 - our [YouTube video playlist](#) (part of the Legislative Drafting Office’s [channel](#)),
 - published articles (see below),
 - conference presentations (mostly remotely, but also in person where affordable), including at ICAIL, ReMeP, LVI-FALM, RaC Guild, and at most of the global and regional [conferences of the Commonwealth Association of Legislative Counsel](#),
 - as well as posts on LinkedIn (and initially on Twitter while legal technologists were still there in numbers, followed by a lower profile on Bluesky).

This helps to promote an environment that drives improvements in legislative drafting and its technology. It also raises Jersey’s profile internationally (particularly in the Commonwealth) as a jurisdiction taking the lead in technology and law. This approach has also helped us to attract excellent assistants, interns and volunteers to our [team](#), whose contributions have boosted our work beyond what was paid for, and who will continue the work into the future.

	Newsletter
	<i>March 2023</i>
Jersey roundup	12
Computer-Readable Legislation Project in Jersey	13
Trainee legislative drafters (“born digital”) in Jersey	14

Our most influential publications, cited by researchers in their papers, have been –

- [Rules as Code: Research Note](#), 2020, Law in Context – 34 citations (Google Scholar) (2,619 reads on ResearchGate) – also referenced in a leading introductory textbook for law students ([Learning Legal Rules](#), Holland & Webb, 11th ed, 2022, p348)
- [Machine-Consumable Legislation: A Legislative Drafter’s Perspective](#), 2019, The Loophole (Commonwealth Association of Legislative Counsel) – 18 citations (Google Scholar) (987 reads on ResearchGate)
- [From pen to algorithm: optimizing legislation for the future with artificial intelligence](#), 2024, AI & Society – 8 citations (ResearchGate)

Our other publications during the project include –

- [Issues with the use and coding of “may” in Commonwealth legislation: a legislative drafter’s perspective](#), 2025
- [Digitizing Legislation: Progress and Prospects](#), 2025
- [Jersey’s project on parsing drafts for if-then structures for “Rules as Code”](#), The Loophole 2023 (2)

These built on our publications from the Legislative Drafting Office’s work on Rules as Code before the project started, such as –

- [Rules As Code: Drawing Out the Logic of Legislation for Drafters and Computers](#), 2022
- [The statute data swamp: NLP, data cleansing, data interpretation and legislative drafting](#), 2022
- [The basics of symbolic formal logic as a useful tool for legislative counsel](#), The Loophole 2021 (2)

Jersey’s work on technology for legislative drafting has also featured in almost every issue of the [Newsletter of the Commonwealth Association of Legislative Counsel](#) since the project began.

7.2 Parsing examples published

Alcohol (Imaginary Provisions) Regulations

THE STATES make these Regulations under Article [42] of the [Tourism \(Jersey\) Law 1948](#)

1 Interpretation

- (1) In these Regulations –
“alcohol” means liquor as defined in the Customs and Excise (Jersey) Law 1999;
“commercial operator” means a body corporate that engages in business for profit but is not a public house or hotel;
“enforcement officer” means person appointed under Regulation [19];
“fraud” includes providing (whether intentionally or not) misleading information in relation to an application for a license under an enactment.
- (2) ...

2 Sale of alcohol

- (1) A person must not sell alcohol, if —
(a) the person is a commercial operator; and
(b) the person has —
(i) an unspent conviction for fraud, or
(ii) an alcohol banning order.
- (2) A person who contravenes paragraph (1) commits an offence and is liable to imprisonment for a term of 1 year and to a fine.

3 Misleading price lists

- (1) An enforcement officer may issue a warning to the proprietor of premises if —
(a) a price list for alcohol is displayed on the premises;

We have published [our parsing set](#) – imaginary and real legislative provisions that illustrate typical structures that need to be captured for computer-readable drafting, including obligations, offences and definitions. We have also published illustrations of different approaches that can be taken to capturing those structures, using –

- [L4](#),
- [if-then statements](#),
- [Excel spreadsheets](#),
- [QnA Markup](#),
- [Mermaid Live Editor](#),
- [DataLex](#) and
- other freely available systems that do not need to be downloaded.

We have not yet managed to render the parsing set in [Blawx](#) ([used by the Canadian government](#)), including a legislative drafter in the federal drafting office), but we hope to do so in future.

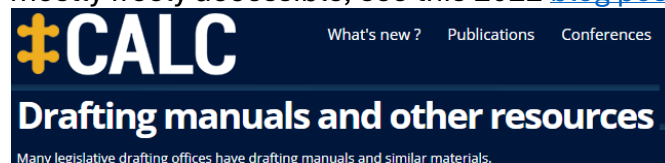
We were able to work with [AustLII](#) on their [DataLex](#) system, and helped them with problems we found with negation of “must” provisions, and with provisions in which there is only a single condition for the effect.

8 Background

8.1 What is legislative drafting?

The [Legislative Drafting Office](#) (or “LDO”) is part of the States Greffe, which serves the [States Assembly](#) (Jersey’s parliament). We are a team of lawyers and legislation editors who draft and publish all of Jersey’s legislation. Article 19 of the [Legislation \(Jersey\) Law 2021](#) sets out the primary objective of the Legislative Drafting Office as “to produce high-quality legislation that can be easily understood and is readily accessible to all persons”. Article 20 of that Law requires the Principal Legislative Drafter to take steps “to promote access to Jersey legislation, including ensuring that it is clearly and logically organised, easy to understand, certain in its effect and as far as practicable conforms with current drafting practice”.

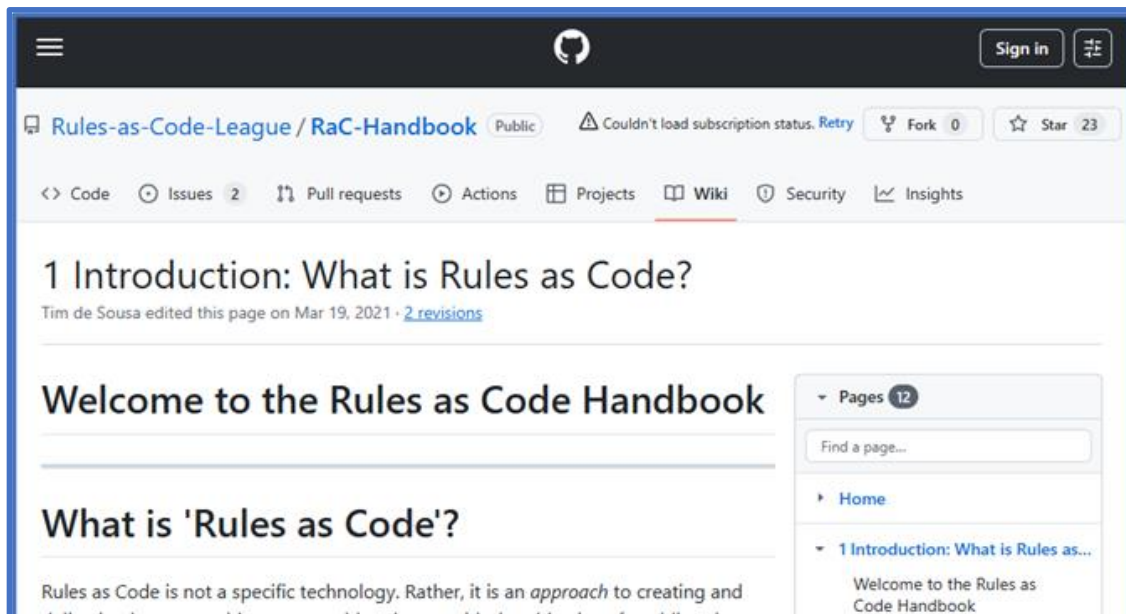
Legislative drafters do not make up the policy that is embodied in legislation. We are given the policy in “drafting instructions” from Ministers, developed by civil servants. Occasionally we draft for backbench Assembly members, who are helped by a team in the States Greffe to develop their policy instructions for us. What the drafter then does is to take the parts of the policy that require a legislative change and express those in a way that is legally workable and fits in with the rest of the statute book (all of Jersey’s current legislation) and Jersey’s customary law. In recent decades we have taken care to draft in plain English, but the current statute book still contains legislation drafted long ago when the style was obscure. Even with plain English, the legislation can only be as simple as the policy that it gives legal effect to. We draft in the modern Commonwealth style, as set out in the [Commonwealth Legislative Drafting Manual](#), but adapted for Jersey as set out in our [Current Drafting Practice](#). We are also a member of the [Commonwealth Association of Legislative Counsel \(CALC\)](#), which promotes cooperation in legislative drafting and training. For links to more information on legislative drafting, mostly freely accessible, see this 2022 [blog post](#).



Our editorial and technology team publishes Jersey legislation through the [Jersey Legal Information Board](#). This means that all current Jersey law is available to the public free of charge, updated as soon as it is amended, as part of JLIB’s commitment as a member of the [Free Access to Law Movement](#). The Legislative Drafting Office is a leader in the use of technology (particularly in publishing continually up to date official versions of legislation). But we, along with all other drafting offices, have inherited problems from the days of paper. One of the main problems is that the reader of a piece of legislation is not pointed to the definitions in that legislation, nor to the definitions and other principles in the [Interpretation \(Jersey\) Law 1954](#) and elsewhere. We rely on those definitions and principles when deciding how to express something in legislation, and our readers need to be aware of them. So the Computer-Readable Legislation Project has been working on helping readers with that (more detail [above](#)).

For more on this, see “[What is legislative drafting?](#)” on CRLP’s website.

8.2 “Rules as Code”



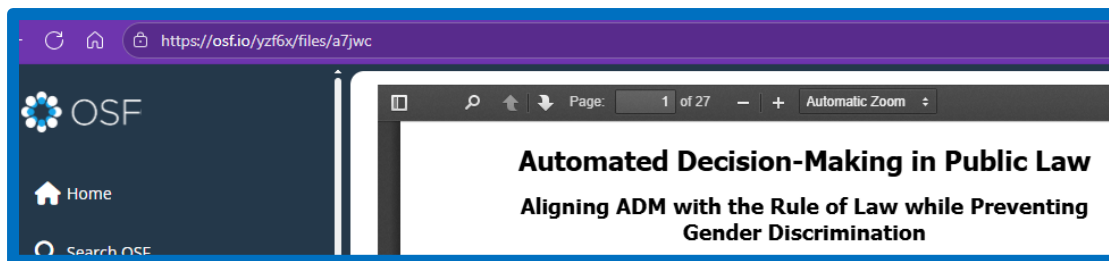
The “Rules as Code” (RaC) movement started in 2018 in [New Zealand](#) as a spin-off from their “[Better Rules](#)” initiative. According to the [RaC handbook](#), RaC is about “drafting and publishing rules in legislation, regulation, and policy in both human and machine-consumable languages (code) so they can be read and used by computers”, ideally through “simultaneous co-drafting of the human-readable and machine-readable versions of the ruleset, to enable the alignment of the intent of policy drafters and the logical constraints of coding” so that the “two versions can then be published together”. According to a 2020 [OECD report](#), RaC proposes rendering legislation “in a machine-consumable form, which allows rules to be understood and actioned by computer systems in a consistent way”. The report calls RaC “more than simply a technocratic solution” and says “RaC represents a transformational shift in how governments create rules, and how third parties consume them”. A [2025 report](#) to the G7 recommends “RaC should be integrated into the rule-drafting process from the outset, with G7-backed pilot projects”.

The RaC movement spread to [Australia](#), [Canada](#) (and see this [video](#)), [Singapore](#), [UK](#), [France](#) and elsewhere – see the 2024 OECD report “[New Techniques for Building and Using Legal Encodings in the Drafting Room](#)”. Jersey’s Legislative Drafting Office became aware of this initiative in 2019 and started work on the idea from our perspective straight away (see [our 2019 report](#) to the Commonwealth Association of Legislative Counsel). Drafting offices in Canada and UK have more recently become involved, but Jersey’s Legislative Drafting Office retains its role as the only organisation consistently contributing the perspective of legislative drafters on how legislation is actually constructed. That perspective is crucial both to keeping RaC within constitutional limits and to giving it the chance to produce results in the real world. That work led to the setting up of our Computer-Readable Legislation Project in 2023.

For more on our Computer-Readable Legislation Project’s approach to RaC and how it differs from others, see our [RaC page](#) on our website.

8.3 Guided understanding, rather than automated decision-making (ADM)

One of the insights we have brought from legislative drafting to Rules as Code is that an over-ambitious version of RaC can be misguided and dangerous to the “rule of law”, but also that a less ambitious version can still bring great benefits (see our [article](#)). In particular we see the benefit of RaC as being mainly in helping readers to follow the logical structure of legislation and in enabling automation of the process of taking human decision-makers to the correct questions that the humans need to decide for a given scenario. We would be worried by versions of RaC that say they aim to automate the implementation of legislation, without caveats, particularly when that is about automated decision-making (ADM) by public authorities under legislation. One of our assistants has written a [paper on the dangers of ADM in public law](#) – the detail reflects her own views, but we agree with the general principles.



We see this as being linked to the question, debated early on in RaC, of what would be the status of the coded version of a piece of legislation. Our view is that legislators are not going to enact code as law, so the coded version is going to have a lesser status than the English text, and it will remain for the courts to decide what legislation means. But there are already things like “Explanatory Notes”, which in Jersey are written by the Legislative Drafting Office, which carry weight and can be admitted as evidence in some circumstances. We then see a layering of roles and versions –

- Drafting offices should produce and publish a basic version of the draft (including the final enacted version), which just tries to capture the logical structure of what the legislation says on its face, without adding material. That could be as simple as an if-then version of the sort that could be used in our [visualiser](#). It could be more sophisticated as long as it still does not go beyond what is on the face of the legislation, the principles that legislative drafters rely on in drafting, and any other similarly encoded legislation. That means it still cannot be used to conclusively determine an individual’s legal position (which needs reference to other law, not just the piece of legislation).
- If a government department administers the legislation then it might produce a fuller encoding that is of more practical use (like the text explanations they currently put out), by adding material that goes beyond what the legislation says on its face (but accepting the risk that a court might disagree). The department might then be able to publish the basis on which its IT system has been set up, assuming that it has not overstepped the mark in letting a computer decide any of the questions that the law requires a human to decide.

- Private sector developers, and non-governmental advice agencies and others, could then build on those versions to produce their own much more fully coded versions, which could automate procedures for their clients. That would be designed to be directly useful to their clients, by incorporating more assumptions (but again with a further increased risk of a court disagreeing).

An example would be sanctions and money-laundering legislation about what banks must do to screen potential customers. The drafting office would publish the basic encoding of the logical structure of the legislation itself. The government department or regulator would publish a deeper encoding that captured its view of the legislation, as expressed in its guidance. A developer would use those to produce software for a bank, which would also incorporate the bank's own policies and systems, and would be used by bank staff to ensure their screening met the legal requirements, the guidance, and the bank's policies.

8.4 Improvements in new legislation, rather than trying to go back over the existing statute book

Although Rules as Code started by being about producing new legislation, much of the focus has since drifted onto encoding existing legislation. If a whole statute book could be automatically and reliably encoded then versions of RaC that deal with existing legislation would become attractive (it would probably also need all case-law to be encoded as well, but that is unrealistic and probably undesirable). But one of the many problems with that idea is that whenever you encode legislation you find logical mistakes which only a court can now correct (unless the legislator drafts and enacts an amendment). Also, no jurisdiction has the resources to encode their whole statute book by using experts, so the process would only be practical if it was automated. If AI was involved, that would increase the risk that legal effects would be changed in ways that were extremely hard to trace.

The advantage of encoding while drafting, instead, is that you can then fix the mistakes before they are enacted. But, as with the point above about depth of encoding, there is a trade-off which is that the statute book will only be coded incrementally as completely new legislation is passed.

In Jersey it is easier than in many other countries to get agreement to update legislation without rehearsing all the arguments about the policy again. Under Articles 14 and 15 our [Legislation Law](#), we also have broader powers than many other countries to correct errors in enacted legislation without having to pass amending legislation. So it might be easier for us to exploit our agility in making an incremental approach move faster, re-enacting a whole Law with a coded version instead of just making substantial amendments to it without any encoding.

As a result of the Computer-Readable Legislation Project, we have also looked at improving existing legislation that creates criminal offences, partly from the work we have been doing to parse offences, and partly from the work on training AI to find offence creation provisions.

- That has led to exposing cases of drafting errors where there is only one solution, where it might be possible to use the correction powers under our Legislation Law.
- It has also led to tightening up our approach to our current drafting to ensure we avoid making those mistakes in future.

8.5 A note for coders on logic programming

We were won over to [logic programming](#) approaches very early on because they reflect what legislation is actually doing. Many programmers coming to Rules as Code want to produce traditional programming and assume legislation is all about mandating processes. So they reckon we just need to automate the process, and then humans cannot break the law. But that misses the point.

- If you can automate the enforcement of the rule then you don't need the legislation. So there is no need for a law to criminalise speeding if you instead have an ordinary English text law to mandate un-hackable speed limiters (with feeds for current speed limits) in all vehicles.
- You only need to legislate when you want to create a legal effect that you hope will help achieve your desired real world effects. So legislation is declarative - “this is the legal effect when these conditions are met (and if there is any argument a court, not a computer, will decide whether the conditions are met)”.

Logic programming uses a [declaratory](#) model which suits this declarative aspect of legislation. The classic logic programming language is [Prolog](#), and logic programming has been recognised as a promising approach for legislation since the 1980s when [Bob Kowalski](#) and others [applied Prolog to the British Nationality Act](#)’s section on citizenship by birth (we have also [used that Act](#)). We have helped to explain to developers the features of legislation that mean it fits the declaratory model, and how that model can provide a reliable backbone for other work with artificial intelligence and with more common [imperative](#) programming languages like [Python](#).

Legislation as Logic Programs*

Robert A. Kowalski

Department of Computing

Imperial College of Science, Technology and Medicine

January 1991

1 Introduction

The characteristic feature of the language of legislation is that it uses natural language to express general rules, in order to regulate human affairs. To be effective for this purpose, it needs to be more precise than ordinary language and, as much as possible, it needs to be understood by different people in the same way. In this respect legislation can be viewed as programs expressed in human language to be executed by humans rather than by computers.

Thus the language of legislation might also serve as a model for computing, suggesting ways in which programming languages might be made more like human languages, while still remaining machine executable. In this paper I shall focus on a comparison between the language of legislation and the language of logic programming. I shall argue that, although logic programming fares well in this

8.6 Key concepts and agencies in this field

- [Artificial intelligence](#) (or “AI”) – not a sharply defined concept, but familiar to most people since the launch of ChatGPT in late 2022. It includes LLMs but also machine learning, symbolic AI, neuro-symbolic AI and other forms. See [4.1](#).
- [Automated decision-making](#) (or “ADM”) – using computers to replace human decision-making, whether by AI or traditional computing. See [8.3](#).
- [Common Legislative Solutions](#) (or “CLS”) – guidance for UK policymakers (being adapted for Jersey). It suggests paths that can be taken on typical issues that come up in many legislative drafting proposals. See [6.1](#).
- [Commonwealth Association of Legislative Counsel](#) (or “CALC”) – the professional body for legislative drafters in the countries of the Commonwealth. Despite our different legal systems, CALC member offices (including the Legislative Drafting Office) share a common modern approach to legislative drafting (see the [Commonwealth Legislative Drafting Manual](#)). See [8.1](#).
- [Jersey Legal Information Board](#) (or “JLIB”) – the independent statutory body on whose [website](#) the Legislative Drafting Office publishes our legislation. See [8.1](#).
- [L4](#) – the domain specific programming language for law developed by Singapore Management University’s [Centre for Digital Law](#) and [Legalese](#). A basic version of L4 is used in the [visualiser](#). See [2](#).
- [Large language model](#) (or “LLM”) – the type of AI that powers chatbots like those from ChatGPT. It is trained on large sets of documents and websites to be able to generate predicted text in response to prompts. See [4.1](#).
- [Legislative Drafting Office](#) (or “LDO”) – the office that produced this report. See [8.1](#) (not to be confused with “LOD”, the Law Officers’ Department).
- [Rules as Code](#) (or “RaC”) – the movement that started in 2018 in New Zealand and sparked the Legislative Drafting Office’s interest in digitising legislation. See [8.2](#).

9 Conclusion

The Computer-Readable Legislation Project has been, and continues to be, a driver for improvement in legislative drafting and for efforts to harness technology to help readers of legislation to find their way through its structures. In the coming years AI is likely to help boost these efforts, but it needs the work of the Computer-Readable Legislation Project to build on. Financial and staffing resources are currently under pressure in Jersey (and many other jurisdictions), but we hope the results from this project will inspire decision-makers to make progress on the opportunities the project has highlighted for Jersey.

Our project's work has helped the ongoing efforts of Jersey's Legislative Drafting Office to improve the quality of drafting and the readability of legislation. But we have also ensured the outcomes of this project have been brought to the attention of other legislative drafting offices in the Commonwealth Association of Legislative Counsel, and amongst the technology community working on "Rules as Code" across the globe. Our work has been recognised as world-leading, and if any countries are inspired to make progress as a result, then Jersey (and other territories or countries) would find it easier to piggy-back on their developments.

Jersey's Legislative Drafting Office will continue to be a forward-looking world leader, punching above its weight and enhancing the reputation of Jersey in the Commonwealth and the wider world.

