Computer-Readable Legislation Project ("Rules as Code")

Legislative Drafting Office, Jersey (see Twitter, LinkedIn & YouTube)

Work Plan - *Initial version as at February 2023*

Contents

1 Outline

1.1 Approach

At Jersey's **Legislative Drafting Office** (LDO) we have just started our 2-year **Computer-Readable Legislation Project** (part of the global "Rules as Code" initiative). It is about enabling legislative drafters to mark up the logical structure of draft legislation so computers can help humans navigate it.

Broadly speaking, our approach will be opportunistic and pragmatic, rather than predicting what we will achieve by when. So we will see which of the lines set out below prove fruitful at any given time, and pursue them to gain the maximum benefit. This version of this plan indicates the initial avenues we will seek to explore.

1.2 Resources

Thanks to the Government Plan we now have some dedicated staff time, so that work on this is not always elbowed aside by urgent drafting. One member of staff (Matthew Waddington matthew.waddington@gov.je) is working solely on the project (half-time, generally mornings), with 3 others already involved to a lesser extent, and with hopes of trying out our approach on the rest of the drafters and editorial staff in the LDO.

We are starting by looking at legislative drafters parsing the logical structure of our drafts, so we are not yet needing tech input. But we do want to establish contact with Jersey's tech community sooner rather than later through Digital Jersey. Where we do need tech input we will start by looking for **free** tech help –

- Existing international contacts (particularly AustLII for DataLex)

- Possibly local law students and computing students – to ask Institute of Law, Highlands College computing and Digital Jersey Academy
- Possibly hackathons – to ask Digital Jersey
- Interested local developers – either attracted to open source elements or interested in getting ahead on developing profit-making tools to use the open source material – to ask Digital Jersey.

1.3 Focus

Main focus – parsing for drafters – how drafters can parse & mark up drafts in a usable way
Second focus – encouraging development of IT tools to use that parsing
Third focus – ways to parse existing legislation (re-enactment, human-reviewed AI/NLP/etc)

Also at this stage we need to produce **introductory material**, & re-connect with contacts in Jersey & overseas – & then use "**working in the open**" to keep in touch with others working on "Rules as Code".

Then there are **related areas** which are not Computer-Readable Legislation or "Rules as Code" as such, but where it may be worth providing some input from our CRLP team because of potential trade-off benefits (such as whether Jersey LDO switches to an XML editor tool for drafting, or produces a Jersey version of Common Legislative Solutions, and so on).

Each of these is set out in some more detail in the rest of this plan.

2 Introductory sessions & material, and working in the open

2.1 Introductory sessions

- Do a **session introducing** the project for all of LDO.
- Demonstrate DataLex, and Excel for Charities Law, show Oracle Intelligent Adviser webpages, explain our parsing project.
- Offer intro sessions for others – LOD, SPPP, M&D, JLIB, students, DJ, JFSC – others?

2.2 Introductory material

- Need to add info about the project on LDO's GoJ website page, then publicise.
- If & when feasible, also make a short intro **video** to which GoJ page can link (hosted on our YouTube, flagged on our Twitter & LinkedIn)
    - Content similar to introductory sessions. Hopping through DataLex, Oracle Intelligent Advisor and Grimmelmann, mentioning our parsing paper for CALC, showing JFSC Regtech information.
    - Either just by recording in Teams/PowerPoint, or by seeking help to make something more interesting (maybe do that later to replace the initial basic one).

2.3 Write-ups & "working in the open"

- Use Twitter, LinkedIn & YouTube to report publicly periodically on our progress (not more than monthly, maybe 6 times a year, when something to say), possibly set up a blog – to be investigated.
- Look at ways to keep interested CALC members informed (Commonwealth Association of Legislative Counsel).
- Write up CALC Loophole article from CALC conference last year.
- Propose report for CALC Europe conference in Cardiff in May – contact UK OPC again about their project. Check other CALC events.
- Investigate POPL-ProLaLa 2024 in London (deadlines this summer). Investigate other conferences (DEON, AUSCL, ReMeP, Bussola, etc), to use as deadlines for reporting on progress through presentations/papers, as well as to disseminate info about our work and invite helpful comment.

3 Main focus – parsing for drafters

Finding a way drafters can be comfortable exposing the logical structure of what they are drafting.

- Take parsing work forward – expose the work to all LDO drafters for any input from them.
- Follow up on demo for idea about copy-paste text into "If-then" boxes with "Is it the case that".

3.1 Elements

- This involves 2 elements –
    - being happy the drafters agree on & understand what the structural elements are & which ones need to be flagged, and
    - being happy that they can do the flagging in a practical way.
- There is then a separate question that needs balancing against the drafter's ease of use – which is the programmer's/computer's ease of use.
    - So what the drafter produces needs to be usable in coded renditions of the legislation (or at least renditions of its logical structure), and preferably usable relatively easily.
    - Ideally the drafter's product would be automatically convertible into coded renditions.
    - But even manually converting it would be an improvement on the current situation.

3.2 Approach

This should just build on the way that drafters already structure their drafts.

- That is by using division into Articles, paragraphs & sub-paragraphs.
- Drafters expose that structure by flagging it (usually with the help of Word styles or XML elements) through numbering, layout, indentation & punctuation that have agreed specialist meanings in legislative drafts.
- For instance drafters (but not enough other people) know the convention that in a paragraphed list the "and"/"or" only comes at the end of the penultimate item, and that you cannot mix "and" & "or" in the same level of paragraphing. So we nest the list with the other connector in a lower level of paragraphing – "(a) xx; (b) xx - (i) xx, (ii) xx, ***or*** (iii) xx; (c) xx; ***and*** (d) xx" (where the roman numeral elements would be indented one more degree than the lettered elements).
- This is the structure of the provisions, rather than the logical ("if-then"/etc) structure, but it can be the starting point for how we ask drafters to flag the logical structure.

3.3 Logical elements

We have already started work on this – presented at CALC conference 2022

- We are starting by focusing on the "if-then" logical element
- We are including definitions in the "if-then".
- We are also looking at the "and/or" element, and how to combine it with "if-then".
- We plan to move on to the "not" element.
- If that works well we could move on to other elements, including –
    - repeated-but-undefined terms, and
    - adding textual-&-enacted versions of ontologies.

3.4 Tech for marking up the parsing

- For the flagging we will mainly start with tech that drafters already have – such as coloured highlighting in Word, or the use of Word styles.
- But we will also look at relatively low-tech (or "No-Tech") ways to illustrate or demonstrate what we are aiming for.
- One way we have started to look at is to drag & drop chunks of drafted text into set types of boxes in a logic flow-chart (perhaps automatically transposed into Excel).

3.5 Checking our logical elements against other existing models for logic of legislation

Particularly for deontic logic, or at least a way to formalise "must", "must not" & "may"

- Coode – & subsequent drafting manuals on (logical) structure of "the legislative sentence"
- Hohfeld – analysis of logical relations between rights, duties, powers, privileges, etc
- Sartor – analysing obligations, rights, powers, constitutive provisions, etc for computer logic
- Declarative/logic programs – Prolog (& Swish), ASP, Logical English, etc

4 Second focus – encouraging development of IT tools to use parsing

4.1 AustLII – DataLex

- Hoping to put some of Charities Law through DataLex.
- See how their system works, particularly yscript & ylegis, for whether our approach can hook in to theirs.

4.2 Investigate other systems

- Oracle (& Monad)
- NAI
- Catala
- L4 (& their spreadsheet system)
- Blawx
- etc.

4.3 Contact Jersey Financial Services Commission again

- Check who is now leading on their **RegTech project**.
- Check where they are up to with their plans for machine-readable Handbooks & Codes of Practice.
- What parsing & markup are they using, how are they going to make their regulatory publications machine-readable, and can we adopt compatible approaches?

4.4 Re-connect with others in Jersey

- Digital Jersey
- Citizens Advice Jersey
- Institute of Law, Highlands College computing, Digital Jersey Academy
- GoJ – Strategic Policy, Planning & Performance; Modernisation & Digital; Digital Economy
- Law Officers' Department; Jersey Legal Information Board; local law firms

4.5 XML

- See what happens with Jersey Legal Information Board looking into converting our legislation into XML for publishing on their site. Look into whether LDO might switch to an XML writing tool, like UK's Lawmaker for writing our legislation.
- Investigate **LegalRuleML** as markup if looks useful & XML editor for LDO looks likely.
- Main issue so far seems to be about overlapping markup – where "if-then" and "and/or" structures overlap; also where we want to flag definitions & offences but they contain (or are contained within) "if-then" structures.

5 Third focus – ways to parse &/or mark up existing legislation (not drafts)

5.1 Re-enactment

- Incremental approach, rather than whole statute book at once. Re-enactment might be attractive once a marked-up piece of relatively self-contained new draft legislation has shown the advantages of publishing the logical structure. Then policy officials and Ministers might want to expand the mark-up into related existing legislation, so as to cover all of a particular field including the existing legislation (as well as what is newly drafted).
- Possibly useful as a way of making progress in financial services, where it could be productive to dovetail with JFSC's work on RegTech (see above) in particular fields.

5.2 Human-reviewed AI/NLP/etc

- See above on **AustLII** for Charities Law in DataLex –**ylegis** for converting existing legislation.
- Tackle legislative copyright and investigate possibility of a new version of open general licence.
- Liaise with **RegGenome**.
- Follow up **Bucerius** Uni (or others from Singapore conf) over using our legislation as **data**.
- Keep an eye on legal users of **ChatGPT** for first draft of parsing/coding existing legislation.

6 Related work

Projects where worth CRLP team having some input, even though project itself is not RaC-driven

6.1 Common Legislative Solutions

- Trainee drafters or students to trawl statute book for Jersey provisions similar to those in the UK versions of Common Legislative Solutions (for drafting instructions).
- Obvious scope to use standardised logical structures alongside standardised provisions.
- Investigate whether students &/or trainee drafters could also produce an app for policy officials to work through CLS when drawing up plans for drafting instructions. Check Flinders University course (see below) for whether their students would be interested.

6.2 Interpretation Law

Work on overhauling the Interpretation Law to make it fit for the digital age.

6.3 "How to Read a Statute"

Fits in with making logical structure easier to navigate, and flagging definitions. Sessions & handout delivered for Jersey's Institute of Law LLB students on reading UK legislation – need to convert it for Jersey legislation.

6.4 Style Manual

Checking that our styles work hand in hand with CRLP approach, & are informed by it, rather than working against it. Feed in to review of LDO Drafting Practice Manual.

6.5 Flinders University projects

Two of our team used this course to develop our award-winning app for policy officers to give us instructions for legislation for annual fee increases. We could now invite their students to work on other projects for us, including Common Legislative Solutions (see above). Could we find students interested in other projects, such as on amending legislation, or other future projects with CRLP relevance?

6.6 Other?