```
+ ---
+ - name: Test LOPE for person in last day of continous training since their 18th birthday
+   period: 2020-09
+   absolute_error_margin: 0
+   input:
+     persons:
+       Scott:
+         date_of_birth: "2002-08-03"
+         date_of_injury: "2020-08-05"
+         acc_part_2__suffered_personal_injury: true
+         acc__has_cover:
+           "2010-08-03": true
+         incapacity_for_employment__corporation_determination: true
+         acc_part_3__has_lodged_claim: true
+         incapacity_for_employment__by_covered_injury: true
+
+         finish_date_of_full_time_study_training_bridging_18th_birthday: "2020-08-05"
+
+         acc_sched_1__engaged_fulltime_study_or_training: false
+
+         acc__earner: fals
+         acc_sched_2__incapacitated_for_6_months: true
+
+         acc_sched_1__weekly_earnings: 500
+     families:
+       Whanau_tahi:
+         others: Scott
+     titled_properties:
+       whare:
+         others: Scott
+   output:
+     acc__potential_earner:
+     acc_sched_1__lope_eligible:
+       "2020-09-06": true
+     acc_part_2__suffered_personal_injury:
+       "2020-08-06": true
+     acc__has_cover:
+       "2020-08-06": true
+     age:
+       "2020-08-06": 18
+
+ - name: Test LOPE for person injured day after their last day of continous training since their 18th birthday
+   period: 2020-09
+   absolute_error_margin: 0
+   input:
+     persons:
+       Scott:
+         date_of_birth: "2002-08-03"
+         date_of_injury: "2021-08-06"
+         acc_part_2__suffered_personal_injury: true
+         acc__has_cover: true
+         incapacity_for_employment__corporation_determination: true
+         acc_part_3__has_lodged_claim: true
+         incapacity_for_employment__by_covered_injury: true
```

Exploring Machine Consumable Accident Compensation Legislation

Lessons for a structural rewrite of the AC Act and opportunities to make it machine consumable

Accident Compensation Better Rules Discovery Team

July 2019

This report can be attributed to:

Accident Compensation Better Rules Discovery Team

[This report is on behalf of this cross-agency team and is not an official view of any agency involved]

Corresponding author:

Jared Stevenson

Senior Policy Advisor

Accident Compensation Corporation

jared.stevenson@acc.co.nz

Cover image: the image on the cover is of non-validated code for the 'loss of potential earnings' entitlement developed as part of the discovery

1 Contents

Figures in this report

Tables in this report

2 Summary

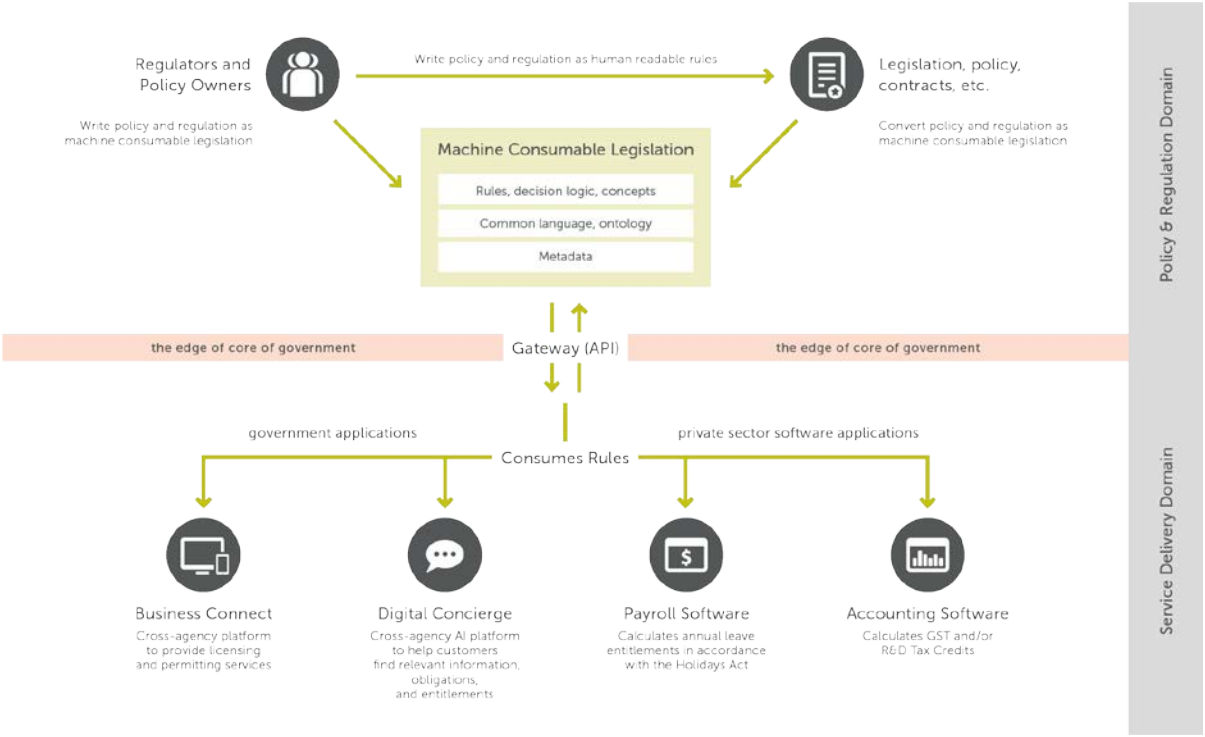
2.1 Introduction

Supported by the Accident Compensation Corporation ('**ACC**'), the Ministry of Business, Innovation and Employment ('**MBIE**') is exploring ways to modernise the Accident Compensation Act 2001 ('**the AC Act**'). This includes exploring a structural rewrite to provide a modern and responsive piece of legislation that supports effective and efficient delivery of the accident compensation scheme.

While exploring opportunities to make the AC Act more accessible and fit for purpose from a structural rewrite, MBIE and ACC learned about 'machine consumable' legislation, and the work of the 'better rules – better outcomes' ('**Better Rules**') initiative and the Service Innovation Lab.

Machine consumable legislation means that its rules can be presented in a structured format (code) that can be processed by a machine (computer) without human intervention. It recognises that machines are ubiquitous, services are increasingly consumed digitally, and there are many uses for machine consumable legislation that are only beginning to be explored. The Better Rules initiative is examining ways to support this through the way New Zealand regulation is developed and implemented, and the Service Innovation Lab previously facilitated a similar experiment.

Figure 1: what a rules repository could look like (Better Rules)



2.2 Approach

The cross-agency and multidisciplinary Accident Compensation Better Rules Discovery Team ('**the team**') undertook a 'discovery' (experiment) over six weeks of half-days based at the Service Innovation Lab starting March 2019. The team comprised legal, policy, legal drafting, business rules, operational, and software development experts from ACC, MBIE, the Parliamentary Counsel Office, the Department of Internal Affairs ('**Internal Affairs**'), the Inland Revenue Department ('**Inland Revenue**'), and the Service Innovation Lab.

The discovery aimed to develop a blueprint of rules for entitlements in the AC Act by developing concept models, decision models, rules statements, and code. The discovery was to support a potential structural rewrite of the AC Act through: creating a shared understanding of the rules for entitlements; exploring use of machines to identify gaps in logic of the rules; providing a practical example that shows how this approach can be applied; and developing tools and techniques that could be applied to future work.

The team recognised that it was unrealistic to complete this blueprint for all entitlements in this time, and developed a prioritised list of entitlements to work through.

2.3 Key findings

a. **Rules statements are of significant value for legal drafters:** Rules statements – a simple set of rules synthesised from complex legislation – that have been robustly developed and tested are expected to assist the Parliamentary Counsel Office (legal drafters) to undertake a more comprehensive and accurate rewrite that is more human and machine consumable.

b. **Developing rules statements identified gaps:** The process of developing rules statements identified gaps in the logic of legislation that had not been previously identified, which range from technical gaps that could be resolved by legal drafters through to gaps that require further analysis (**Lesson 19.**). By resolving these logic gaps, legislation can be more human and machine consumable, the risk of misinterpretation can be reduced, and what is implemented can better match policy intent.

c. **Legislation may have longer rules and more of them:** A prevailing view in the team at the beginning of the discovery was that a structural rewrite would result in fewer or shorter rules to make the AC Act more accessible and human consumable. To the contrary, the team found that ensuring there are no gaps in the rule logic can result in longer rules. However, these longer rules were clearer and would ensure legislation is more human and machine consumable. Furthermore, as legislation becomes machine consumable, and policy developers have tools to better understand and model its (positive and negative) effects, additional rules may be created to create more nuanced policy.

d. **Developing rules statements and code at the same time most effective:** The discovery initially explored handing rules statements to software developers for turning into code. However, it was found that writing code at the same time as rules statements produced better quality rules statements and code, and was more efficient. It improved shared understanding, reduced follow up questions, and allowed the code to validate the rules statements.

e. **Concept models improved shared understanding:** The use of concept models as the first step for any new topic improved shared understanding and team engagement. They showed not only what was in scope but also what was out and enabled focused and effective communication. Concept models proved to be a core first step for the multi-disciplinary team to start working on each entitlement.

f. **A small multidisciplinary team can produce most outputs:** While the discovery had a broader team to draw on (and who were all used), the outputs that would support a structural rewrite were produced with a small team (fewer than six) of experts across business rules, operations, policy, legal, legal drafters, and software development.

g. **Tools increased efficiency and broadened use of code:** During the discovery, the team identified areas where different tools may have been of value, including tools for capturing concept models, tools that can test policy, and tools for storing rules in a dataset form from which executable legislation as code could be derived. The team experimented with tools and found that they demonstrated value and (at least in one case) were complementary to each other, but neither is necessary for a structural rewrite alone.

h. **Rules requiring human judgement can be written in code:** The team explored developing code for an area of legislation requiring significant human judgement to test whether it is difficult to code, and found it was no different or more complex than other rules. It emerged that identifying these types of rules could be used to track the quantity of human judgement within a piece of legislation, which could provide a new perspective and way to measure how much a piece of legislation introduces.

i. **Value of software developers is beyond code:** The software developers brought a lot of value to the multi-disciplinary team that was beyond the code they create. They looked at rules and logic on behalf of the machine, and improved the logic and robustness of rules statements.

j. **A multidisciplinary team improves implementation:** Using a multidisciplinary team to develop rules statements improved connections of people involved in policy development through to those in operations, and was observed to reduce translation gaps and is expected to improve implementation. It also sped up the cadence of work as the experts were in the room at the right time, eliminating handover errors and delays that usually occur when experts from a different team provide input.

2.4 Opportunities

a. **Develop rules statements for other parts of the AC Act:** To support legal drafters to ensure a structural rewrite would result in more human and machine consumable legislation, MBIE and ACC should develop rules statements for all provisions in the AC Act that may benefit from this work. A pragmatic approach would be to develop a prioritised list of provisions that would benefit most from rules statements, and scope the effort (time and resources) to complete that work in the time available. This does not require another intensive discovery, but rather could use regular discrete blocks of time (eg, 4 hours per week over six months). To develop code that tests the logic, a software developer is required.

b. **Use methods from the discovery for other policy and legislative changes:** While the focus has been on a structural rewrite, the approach tested across the six-week discovery could also be used as part of everyday business when considering policy and legislative changes. While this is particularly true for the AC Act, which is complex and composed of many interrelated provisions, considering different policy and legislative proposals against rules statements and code would robustly test their logic and implications.

c. **Improve macro view through concept models:** MBIE and ACC should seek opportunities to improve the macro view of accident compensation scheme settings through concept models. This could give a transparent and consistent view of the accident compensation scheme, and provide a common foundation for cross-agency collaboration, in the same way the smaller and more targeted concept models did for the discovery.

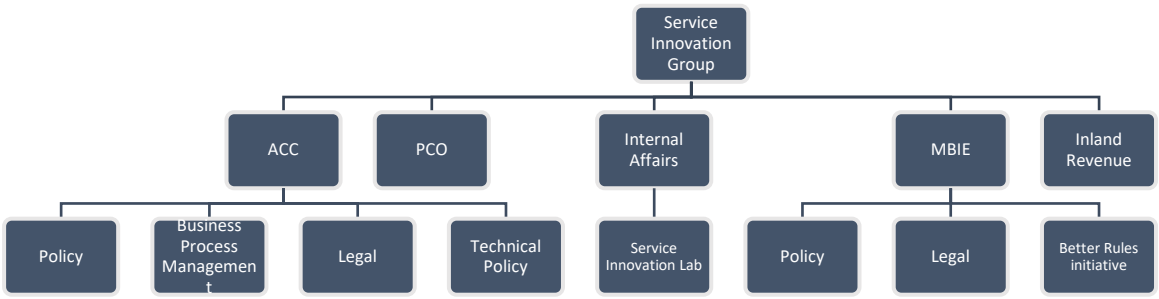
d. **Code the whole AC Act and make it public:** To fully support the long-term ambition of machine consumable legislation in New Zealand, ACC, MBIE, the Better Rules initiative, and the Service Innovation Lab could explore coding the whole AC Act. If opportunity **a.** above happens, much of the AC Act would have been coded anyway as part of that process. A key realisation of the potential of machine consumable legislation is publicly publishing the code in an executable form for the same reasoning governments publish their legislative text: to aid people's comprehension of the law. Coding the AC Act would be dependent on the Better Rules initiative completing its planned work on developing New Zealand standards and architecture for machine consumable legislation.

e. **Continue to explore using code already developed:** Some code has already been developed as part of the process to develop rules statements (and more would be developed if opportunity **a.** above happens). This code presents a unique opportunity for innovation. For example, it could be used for policy development (eg, modelling test cases), for consumers or advocates to develop an end-user interface, or for government to see how the accident compensation legislation interacts with other legislation. Working with the Better Rules initiative and the Service Innovation Lab, ACC and MBIE should continue to explore this so that the opportunity is not wasted.

3 Purpose of this report

1. As highlighted in **Figure 2**, there are several New Zealand government agencies and organisations who would be interested in the outcome of the discovery. There is also interest internationally to understand whether the discovery provides new information or supports existing knowledge on machine consumable legislation.
2. To serve the broad range of audiences, this report:
    a. summarises the approach taken and lessons learned from the six-week discovery that explored how to make the AC Act more human and machine consumable
    b. provides MBIE and ACC approaches for making a structural rewrite more human and machine consumable
    c. discusses the value of machine consumable legislation and the Better Rules initiative, providing the Better Rules initiative insights to further develop its approach
    d. provides the Service Innovation Lab a summary of the discovery and how it might support MBIE and ACC goals in the future.

Figure 2: New Zealand stakeholders for the discovery



4 Background

4.1 Legislative modernisation

3. Supported by ACC (the operational agency), MBIE (the administrator of the AC Act) is exploring opportunities to modernise the accident compensation legislation by making a suite of legislative changes, which includes a possible structural rewrite.

4. The intent of a structural rewrite would be to develop a modern and responsive piece of legislation that supports effective and efficient delivery of the accident compensation scheme by:

    a. reordering the AC Act for ease of use and navigability

    b. clarifying terminology and using plain language

    c. removing unnecessary prescription

    d. ensuring consistent delegation of decision-making powers.

5. When this report was written, Cabinet had not approved any structural rewrite. If drafting instructions are issued, a structural rewrite is expected to take the Parliamentary Counsel Office at least 12 months to complete.

4.2 Machine consumable legislation

6. Machine consumable legislation (also referred to as 'machine readable legislation' or 'legislation as code') means that its rules can be presented in a structured format (code) that can be processed by a machine (computer) without human intervention.

Machine readable

Information or data presented in a structured format that can be processed by a computer without (or with minimal) human intervention and without loss of semantic meaning. Digital formats are not automatically machine readable (eg, text documents in PDF or DOC formats are not machine readable). *(OECD, 2016)*

7. The traditional models of creating, using, and improving regulation (rules of government, which includes policy, legislation, and business rules) were developed for use in a non-digital environment. That regulation, however, is already having to be converted to a machine consumable format because of the ubiquitous nature of machines within government agencies, and can result in a mismatch between policy intent and what is implemented. MBIE's Better Rules initiative emerged to better match intent with outcomes, recognising that services are consumed digitally.

8. The Better Rules initiative examines the way the New Zealand government develops and implements regulation to ensure that it can be more easily consumed by digital channels. It involves reimagining regulation as an open platform based on logic, decision models, and rules. For examples of a future state that were developed from a previous experiment, see '**Better Rules for Government Discovery Report**'.

9. The Better Rules initiative ensures New Zealand has a platform that future proofs its approach to regulation in an increasingly digital world. It is about policy-makers, regulators, legislative drafters, service designers, software developers, and impacted people working closely together to clearly articulate the rules underpinning the regulation and how those rules will work in practice.

10. In the future, the machine consumable version of the regulation (code) would be publicly available from a single authoritative source that anyone could use (see **Figure 1**). This would provide an opportunity for government agencies, private organisations, and individuals with new ways to comprehend and work with regulation to design and deliver a myriad of integrated services. For example, services that allow agencies to model (intended and unintended) effects of a proposed policy change, and services that allow citizens to easily engage with matters of law.

11. The Better Rules initiative is about reframing the regulatory design process using an end-to-end system design approach to enable regulation to be more easily implemented as part of government's digital services. It ensures that regulation is developed to be machine consumable, in parallel with the (current) human readable version.

12. The Better Rules initiative is jointly led by MBIE's Better for Business programme and the Service Innovation Lab, and collaborates with New Zealand government agencies, including Inland Revenue and the Parliamentary Counsel Office.

4.3 Service Innovation Lab

13. The Service Innovation Lab is an interdisciplinary team with backgrounds in agile, design, policy, and technology. It uses evidence-based decision making and design thinking to transform government service experiences.

14. The Service Innovation Lab works in the open to prevent problem shifting and duplication and increase the sharing, reuse, and adaptation of tools and approaches.

15. By partnering with other agencies, not for profits, private organisations, and other governments, the Service Innovation Lab can understand the whole service system to identify and test the best ways to create lasting change. Its approach aims to reduce risk and increase the impact of innovation across government.

16. The Service Innovation Lab had previously facilitated a similar experiment to the discovery, exploring the challenges and opportunities of developing human and machine consumable legislation for effective and efficient service delivery (see 'Better Rules for Government Discovery Report').

5 The discovery

5.1 Overview

17. A cross-agency and multidisciplinary team agreed to undertake a discovery over six weeks of half-days based at the Service Innovation Lab, starting 4 March 2019. The discovery was to develop a blueprint of rules for entitlements in the AC Act.

18. The discovery was a convergence of the legislative modernisation project, the Better Rules initiative, and the Service Innovation Lab. The decision to proceed with the discovery followed two workshops with relevant stakeholders in 2018 to better understand what a discovery would entail, and the corresponding plan was outlined in the statement of work, *Supporting ACC Legislative Modernisation* (February 2019).

5.2 Aims

19. The discovery aimed to:

    a. create a shared understanding of the rules for entitlements (and produce related outputs) for other parts of the legislative modernisation process, such as for drafting instructions

    b. use a cross-agency multidisciplinary team to explore how the process of developing legislation from policy can use machine logic to ensure rules are transparent and understandable

    c. identify how other stakeholders could be involved in and contribute to a process like this (eg, claimants, providers)

    d. provide a practical example that shows how this approach can be applied to teams across ACC, MBIE, and other interested agencies

    e. develop tools and techniques that could be applied to future work.

5.3 Scope and outputs

20. The discovery was limited to developing a blueprint of rules for entitlements in the AC Act (Part 4 and Schedule 1). Excluded from this blueprint were the transitional provisions and the impact of case law.

21. A blueprint of the rules for entitlements was to emerge from developing:

    a. **concept models:** used to map, understand, and validate the high-level design of the policy, and to develop shared understanding

    b. **decision models:** used to understand and validate the high-level logic of the policy (these were more of a process step than a key artefact for the discovery)

    c. **rules statements:** used to give effect to and articulate the high-level design and provide detailed understanding of the legislation

    d. **code:** used to test the logic of the rules statements and develop applications that accurately reflect the legislation.

22. The scope of the discovery did not include providing legislative deliverables, providing outputs intended to integrate into ACC operating systems, or changing existing policy processes.

5.4 Discovery plan

23. The discovery was hosted by the Service Innovation Lab, who provided a collaborative space as well as software developers and a 'scrum master'.

24. While the initial plan was to undertake a three-week discovery (120 hours of collaborative time), the core team could not commit full-time. The discovery, therefore, was spread out over six weeks using half days, meeting every morning.

25. The team expected that it would be unrealistic to work through all rules for entitlements in the allocated time, so the team divided the entitlements into manageable chunks and worked through a prioritised list.

26. The intention was to have a regular demo or showcase for respective managers, likely to be at the end of a two-week block.

5.5 The team

27. The discovery was established with a mix of core (Table 1) and support (Table 2) team members, to ensure the core team was focussed, but also to provide adequate expertise as required.

Table 1: core team

Organisation	Specialty	Number of staff	Capacity (average)
ACC	Business rules	3	45 hours per week
	Legal	2	20 hours per week
	Policy	1	30 hours per week
	Product owner	1	15 hours per week
Service Innovation Lab	Better Rules initiative	1	20 hours per week
	Scrum master	1	20 hours per week
	Software developer	2	20 hours per week
Parliamentary Counsel Office	Legal drafter	1	10 hours per week
MBIE	Policy	1	12 hours per week

Table 2: support team

Organisation	Specialty	Number of staff	Capacity (average)
ACC	Technical policy	2	30 hours total
	Technical specialists	3	50 hours total
	Legal	1	4 hours total
MBIE	Better Rules initiative	1	20 hours total
	Service design	1	10 hours total
Inland Revenue	Better Rules initiative	1	20 hours total
	Tool specialists	3	24 hours total
Service Innovation Lab	Software developer	1	20 hours total
SmartCore	Director	1	20 hours total

5.6 Way of working

28. The discovery brought together team members from multiple agencies with diverse specialist skills who were passionate about the potential of the discovery.

29. On coming together, the team defined its way of working, and looked to agile, lean, and innovation frameworks to achieve this. The team wanted to work collaboratively, learn from each other, continuously improve, and experiment with different approaches.

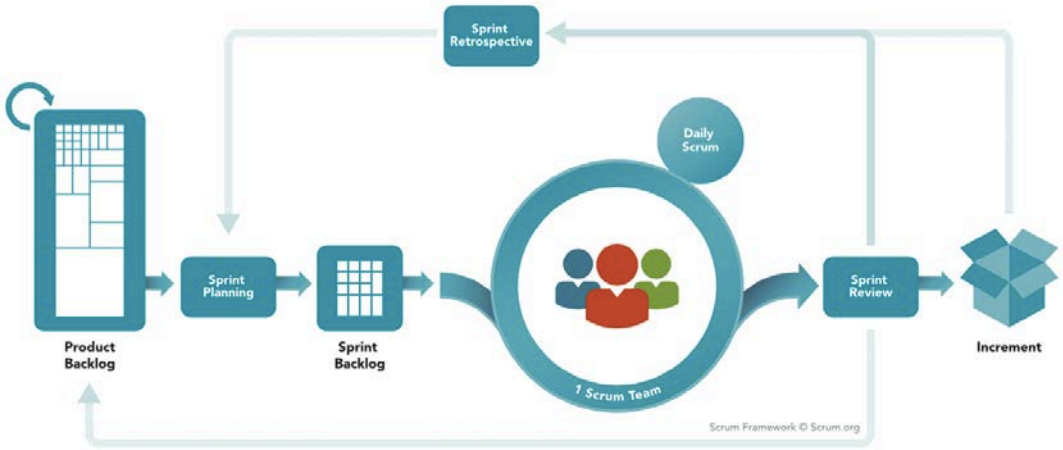
Team charter

30. The team started by creating a team charter. This brought the team together with a common understanding of why the team was there, and how the team would operate and interact. The team agreed to adopt an agile and experimentation mindset, work collaboratively with a high level of trust, and respect the diversity of skills and organisational drivers.

Scrum framework

31. The team adopted the scrum framework. This was new to most team members, meaning there was a period of learning. However, this small investment enabled the team to foster the experimental and continuous improvement mindset, learn quickly, and explore how the framework or elements of the framework would be helpful.

Figure 3: scrum framework

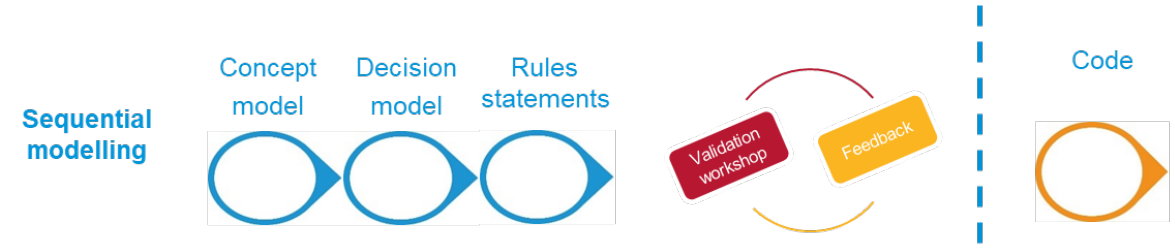


32. Entitlements in the AC Act were divided into an ordered backlog and the top item, the ‘loss of potential earnings’ entitlement, was taken into the first sprint. This item was selected as it was a small finite area with low risk and high opportunity to learn. As backlog items were completed, the next highest value item was taken. The team reflected on the lessons of each sprint and the team’s processes, ceremonies, and way of working evolved as a result.

Sequential modelling

33. The team first explored sequential modelling (see **Figure 4**). This required cross-agency and multidisciplinary collaboration to develop concept models, decision models, and rules statements. The rules statements were then 'handed over' to the software developers for code creation (note the original plan was to create code simultaneously with the rules statements but it did not transpire this way).

Figure 4: sequential modelling



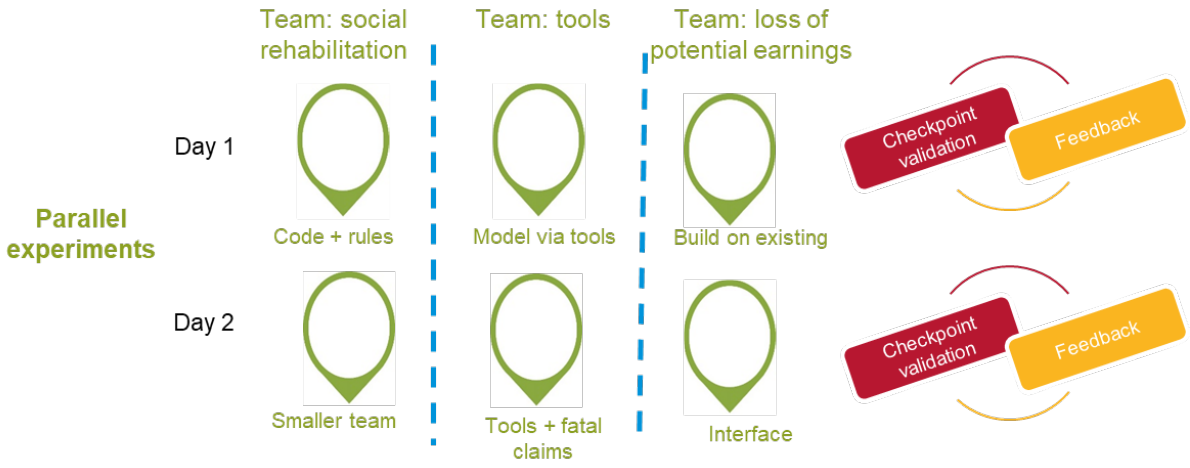
34. This was a quick and easy to follow approach that was used for a few backlog items. One of note is the 'loss of earnings' entitlement – a complex area that was completed in nine hours from concept model through to rules statements. This included identifying and discussing issues and ambiguities, and enabled the team to move quickly and gain new understanding, even for subject matter experts.

35. This approach did have limitations. Handing over rules statements meant software developers had to regularly ask questions to gain sufficient understanding to create code. This was inefficient and could introduce translation gaps, and missed an opportunity for the code to meaningfully test or validate the logic of the rules statements. It highlighted the importance of getting the software developers involved early, which the team adopted in subsequent sprints.

Parallel experiments

36. To make use of the diverse talent and variable availability of team members, the team ran parallel experiments in week five (see **Figure 5**). This involved taking items from the ordered backlog, drafting an experiment canvas based on the problem definition, creating teams with the right skills and expertise, and sprinting for two full days (these days were in the same week, but not consecutive).

Figure 5: parallel experiments



37. During the parallel experiments, all teams came together for checkpoints in which a team representative identified what they had been doing since the last checkpoint, what they were planning to do next, and anything they needed help with. These checkpoints allowed cross-team collaboration and understanding, and let the teams reform where needed, supplemented by other skills.

38. **Table 3** provides an overview of what was done in these week-five experiments, which proved to be an efficient, enjoyable, and valuable way of working. It limited the impact of context shifting outside of the project and made maximum use of the specialist skills inside the team.

39. For further discussion on the tools used, see section **6.6** (**Tools**).

Table 3: parallel experiments conducted in week five

Team	Representation	What they did
Social rehabilitation	Policy, legal, software developers, and business rules	Formed around a new backlog item ('social rehabilitation' entitlement) and explored creating rules statements and code at the same time. Proved to be an informative and effective way of working, even more so on the second day when the team was smaller
Tools	Policy, legal, software developers, business rules, and experts in Oracle Policy Automation and SmartCore	Initially explored the 'loss of earnings' entitlement, which had been created using sequential modelling to see how this might compare using tools. On day two, this was built on by exploring with a new item from the backlog ('fatal claims' entitlement), with more intensive use of legal and business rules expertise
Loss of potential earnings	Software developers, policy, business rules, and Better Rules initiative	Built on (what was then) the most explored entitlement, 'loss of potential earnings', by exploring the use of data and an interface for that data

Collaboration

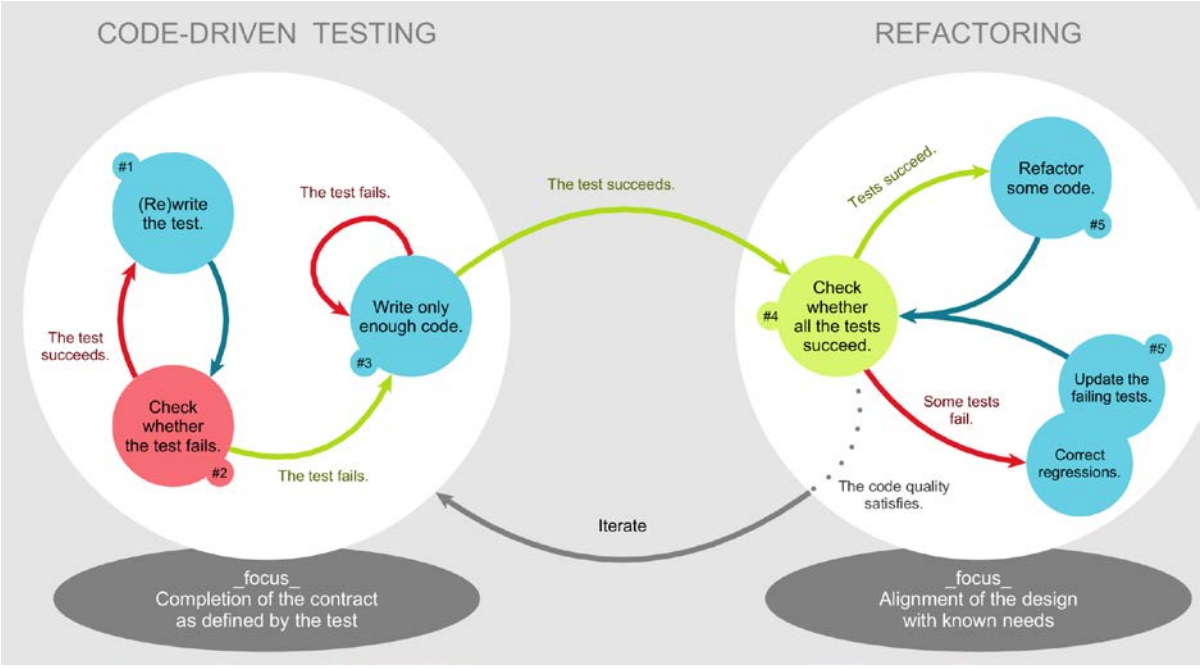
40. Collaboration was at the core of the team's approach and facilitated the positive outcomes and learning of the discovery. It was fostered by physically locating the team in a neutral space separate from their normal business environments.

41. The team considered, but did not use, cloud-based tools for offsite collaboration due to information security concerns. The team used Trello, an online tool that provides a mechanism for recording, assigning, and tracking backlog items. It also became a central repository for project resources.

Test-driven development

42. Test-driven development was also explored during this sprint (see **Figure 6**). A multidisciplinary team creates a test case. The software developer then creates only enough working code to pass that test. When the test succeeds, the next test case is created, and only enough code is created to pass that test, too. As the test suite grows, all tests are run to ensure they all pass with the new changes to code.

Figure 6: test-driven development



6 What was achieved

43. In developing the blueprint of rules for entitlements, the team produced various combinations of outputs.

44. As the team had an experimental approach, the success criteria for artefacts were agreed upfront, to ensure that the team knew when to move on. In some cases, for example, the artefact was a photo of post-it notes, or in other cases the acceptance criteria was whether the artefact was of value to the next stage. Outputs and workings were saved in Trello.

45. The Better Rules initiative focuses on creating knowledge assets that can be shared and re-used by the different rules stakeholders. Their purpose is to create a common knowledge and understanding of an area. The key knowledge assets for this discovery were concept models (**Figure 7**) and rules statements (**Figure 9**).

Table 4: artefacts from the discovery

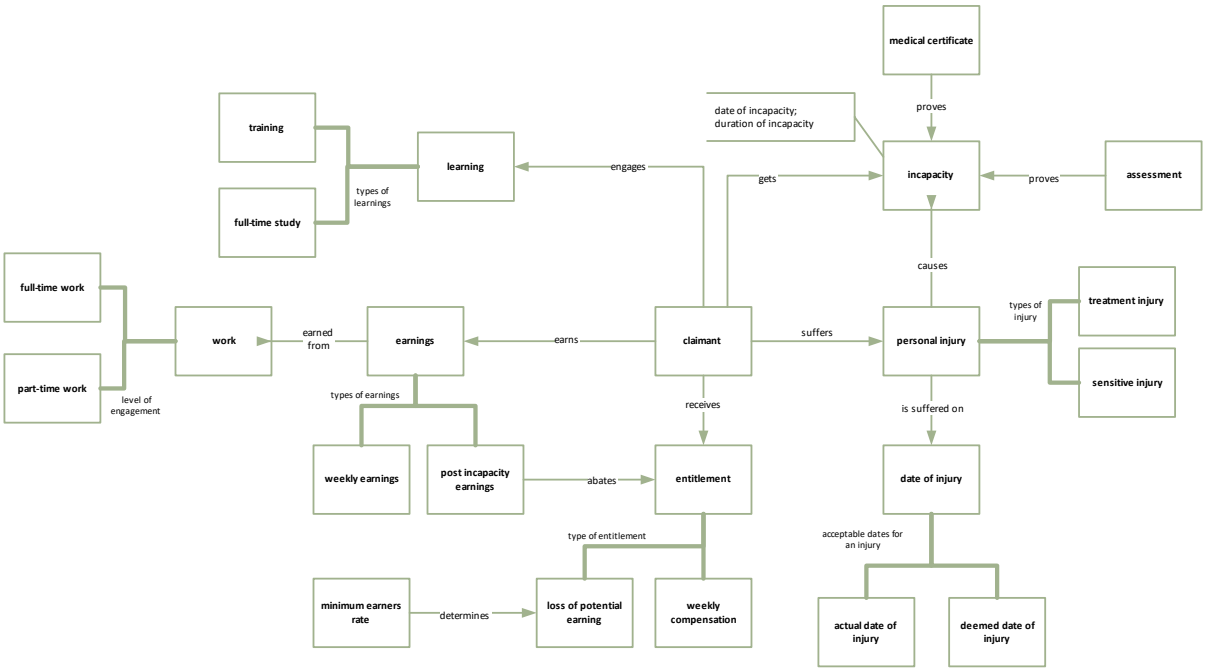
Areas of the AC Act	Concept model	Decision model	Rules statements	Code	Interface	Tools
1. Entitlements – overview						
2. Loss of earnings						
a. Eligibility						
b. Amount						
3. Earnings assessment						
4. Ceasing weekly compensation						
5. Loss of potential earnings						
6. Incapacity						
7. Fatal claims					SmartCore	
8. Social rehabilitation						
9. General claim process						
10. Purpose of the scheme						

6.1 Concept and decision models

46. Concept models (**Figure 7**) focussed on creating a common understanding of the concepts within an area and showed the relationship between concepts, while decision models (**Figure 8**) showed operationalisation of policy through a decision-making approach. The decision model approach fitted well in this experiment due to the nature of the rules the team was looking at. These two assets were used to develop more concise rules statements.

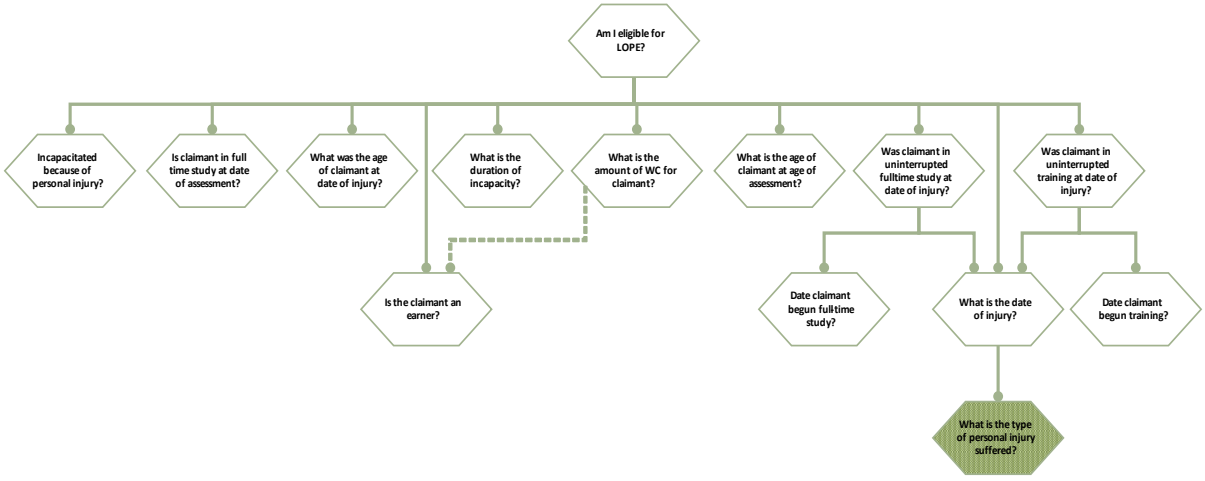
47. The team developed concept models for all entitlement areas the team considered. While few in the team had experience with concept models, they quickly became a critical method to develop a shared understanding. All concept models started as post-it notes on a whiteboard. Some were further defined in Visio, while others were digitised as images of the post-it notes or by other visualisation tools (eg, Draw Express Lite), which enabled visualisations via smartphone to be created in parallel with team discussions and avoid bottlenecks of later creation.

Figure 7: example of concept model ('loss of potential earnings' entitlement)



48. Decision models were a less defined artefact for the discovery, and were a necessary step between developing concept models and rules statements (ie, the value for how the discovery used decision models was more in the process than the artefact).

Figure 8: example of decision model ('loss of potential earnings' entitlement)



6.2 Rules statements

49. Rules statements proved themselves to be a key deliverable and were captured in Word documents (**Figure 9**). These rules statements synthesised the related rules, spread throughout the AC Act, into a simple set of rules. The process of getting these rules (ie, developing concept models, decision models, using software developers, developing code to validate their logic, and using subject matter experts) ensured that the outputs were considered a fair and reliable summary of the rules.

50. Validating the rules statements came to be recognised as a necessary step. This is where additional subject matter experts were engaged to go through the rules statements, line by line. This step added significant value. Validation was not a formal step for the concept models and decision models.

Figure 9: example of rules statements ('ceasing weekly compensation')

Rule Statement	Source
A claimant is no longer entitled to WC if all of the following are true:	S 103, 105
• The claimant is in receipt of WC	
• At least one of the following:	
○ A clinician confirms claimant is able to perform suitable work based on at least one of the following:	
▪ Experience	
▪ Training	
▪ Education	
○ A clinician confirms claimant is able to perform pre-injury work.	
○ Clinician confirms claimant's incapacity is no longer due to the covered personal injury.	s 117(1)
○ Claimant has unreasonably refused to comply with a reasonable requirement.	s 117(3)
A claimant is no longer entitled to WC if all of the following are true:	ss 107 - 113
• The claimant is in receipt of WC	
• The claimant continues to be incapacitated.	
• Claimant has completed vocational rehabilitation.	
• ACC considers vocational independence is likely.	
• A vocational assessor has assessed at least one suitable work type for the claimant based on at least one of the following for the claimant:	
▪ Experience	
▪ Training	
▪ Education	
• A medical assessor has assessed the claimant as fit for at least one of the suitable work types identified by the vocational assessor.	

6.3 Code

51. Code offered three key components to the process. The first is the discipline of having to explain the policy in a way that a non-thinking, non-assuming machine can calculate. It is a process which leverages the machine's complete lack of imagination and assumption to highlight gaps in the logic. This does not always capture unusual scenarios, but a software developer skilled in analysing logic models and writing tests for them will often be able to discover these during the coding process. The writing of the code is partnered with the development of a set of tests with expected outcomes (see **Figure 10**). Whether the developed code is used and published is a separate discussion to the value that just writing the code offers to the policy development process.

Figure 10: example of code ('social rehabilitation' entitlement)

```
...  @@ -0,0 +1,49 @@
1    + # -*- coding: utf-8 -*-
2    +
3    + from openfisca_core.model_api import *
4    + from openfisca_aotearoa.entities import Person
5    +
6    + # (a) # aids and appliances:
7    + # (b) # attendant care:
8    +
9    + class acc__is_entitled_to_attendant_care(Variable):
10   +     value_type = bool
11   +     entity = Person
12   +     definition_period = MONTH
13   +     reference = "Section 81 & 82 http://www.legislation.govt.nz/act/public/2001/0049/latest/DLM101425.htm
14   +
15   +     def formula(persons, period, parameters):
16   +         return persons('acc__has_a_covered_injury', period)
```

52. The second component is the ability to develop a test suite of scenarios that the code can run. Tests are used in code development to ensure the code is delivering the expected outcomes (see section 6.4 [Test suites]). Code offers the ability to store and run quickly thousands of tests in ways that far exceed human thinking capabilities.

53. The third component is using the code to run large and small scale simulations that can use datasets representative of the affected population and quickly provide answers and insights that inform policy development. The machine can parse the complexity much more quickly and accurately than the human brain.

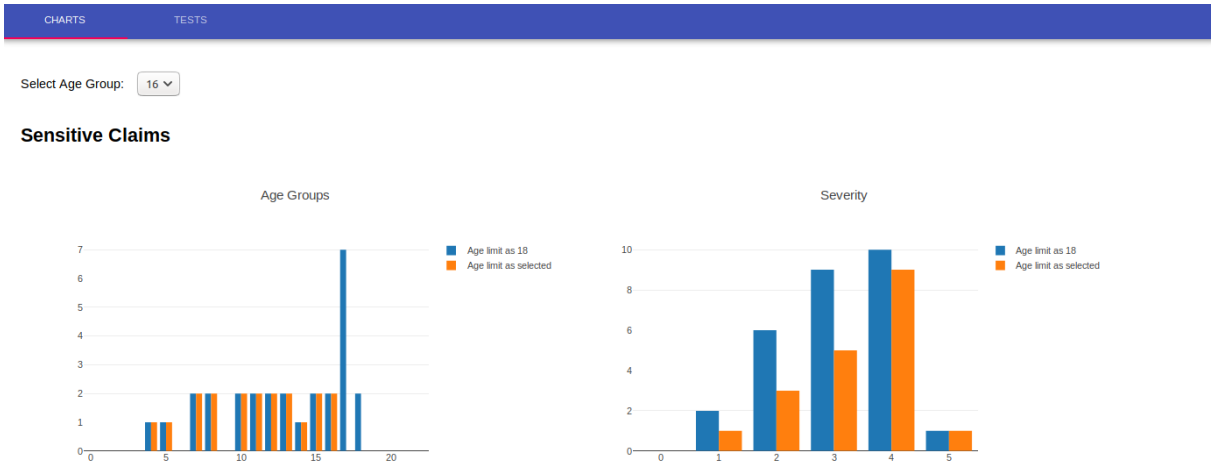
6.4 Test suites

54. Test suites are collections of scenarios with expected results based on policy that the code can execute and test for the expected output. These test suites can live on alongside the code as a way of capturing policy intent.

55. Over time, real world scenarios that come up against the legislation in ways that were missed initially can be captured as tests and added to the suite. This offers an ability to capture and preserve the impact of the policy on people in more unusual situations. These test suites would become invaluable in future changes to policy as they would be able to highlight unintended consequences quickly and efficiently. This would be particularly true for outlier situations that would, with current approaches, be too difficult to address.

6.5 Interface

Figure 11: policy visualisation tool ('loss of potential earnings' with example data)



56. An interface (or dashboard) for the 'loss of potential earnings' entitlement was developed to explore the potential for machine consumable legislation (see Figure 11). This was to provide an example of what could be used for policy development.

57. The interface allowed the user to change the date of entitlement for 'loss of potential earnings' and to see the effects that change had on how many were eligible, the types of injuries they had, and the severity of those injuries. This interface experiment used example data, but could be easily swapped out for real world data.

58. The interface was elementary, but illustrated a future where policy development could use rule sets (machine consumable legislation) to help in researching new opportunities or refinements of existing legislated policy.

6.6 Tools

59. As discussed in section **5.6** (**Way of working**) above, the approach taken for the first few weeks of the discovery was to produce concept models, decision models, and rules statements independently of code, and hand this over to software developers to create code.

60. The process of developing policy, rules statements, and code jointly is an area under active investigation worldwide and the approach taken, which included a multi-disciplinary team, is currently considered the best approach. There are many tools and methods being explored to better capture and improve this process and it is an area likely to see rapid development. The multi-disciplinary team however will likely remain a key component to the labour-intensive process due to the need for human input, understanding, and insight. The Better Rules initiative is currently working on a report focused specifically on the tooling and architectural considerations that affect this work. This report will be published in the second half of 2019.

61. The discovery experimented with three different code tools in week five:

    a. **OpenFisca:** An open source tool developed in France primarily for the publishing of legislation as code to be consumed via an internet based REST API. The Service Innovation Lab has been using this as a tool in its ongoing lab experiments in the Better Rules space.

    b. **Oracle Policy Automation:** A tool used in the government sector, which was demonstrated by Inland Revenue specialists who are using this. Oracle Policy Automation develops and captures rules statements from the decision model, and generates code from the rules statements. The Oracle Policy Automation suite includes interfaces that enable interacting with the code for testing and scenario modelling.

    c. **SmartCore:** A toolset not yet in production (beta version), which was operated and demonstrated by the director of SmartCore. SmartCore enables testing of the models and code through a simple interface (currently the SmartCore inputs and outputs are generated through an Excel-based user interface).

62. The following describes what the team did with the tools:

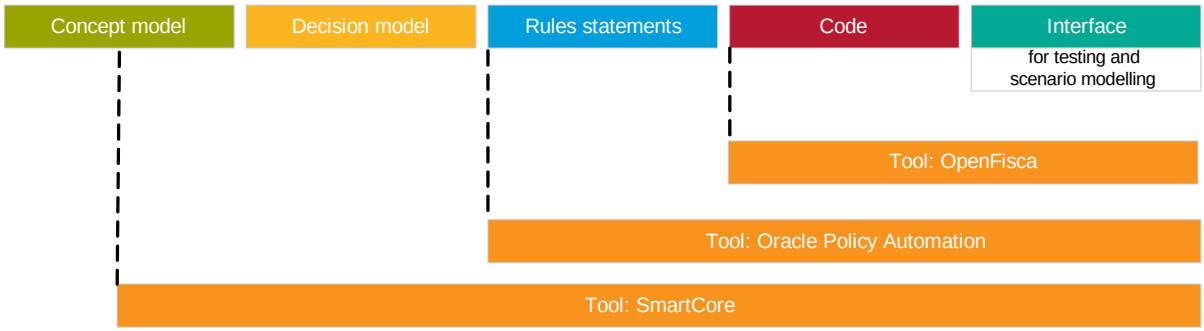
    a. OpenFisca was used to model the ‘loss of earnings’, ‘loss of potential earnings’, and ‘social rehabilitation’ entitlements. The ‘social rehabilitation’ entitlement was chosen because it requires a lot of human judgement decisions.

    b. Oracle Policy Automation and SmartCore worked in conjunction to model the ‘loss of earnings’ entitlement **after** the concept model, decision model, and rules statements were developed.

    c. SmartCore modelled the ‘fatal claims’ entitlement at the same time as the development of the concept model, decision models, and rules statements.

63. Figure 12 shows where the tools interacted in different parts of the process. OpenFisca is primarily a rules engine and currently occupies the code part of the process (along with an experimental interface tool called RapuTure [http://www.rules.nz]).

Figure 12: where in the tools interacted in the discovery's process



Oracle Policy Automation

64. Specialists from Inland Revenue input into Oracle Policy Automation the previously-developed rules statements for the 'loss of earnings' entitlement as natural language code.

65. Using this tool, the specialists recreated the rules statements (Figure 13) and identified gaps in the logic, including the order of the rules. The tool provided an interface for interacting with and testing the rules. Based on the same rules statements, the interface could be amended to use technical language for internal use or use plain language for a user-friendly experience.

66. As examples of outputs from this experiment, Figure 14 shows a data interface, Figure 15 shows the interview screen interface, and Figure 16 shows the test cases interface.

SmartCore

67. The discovery highlighted that tools can be complementary. For instance, the Oracle Policy Automation test cases for the 'loss of potential earnings' entitlement were mailed to and consumed by SmartCore.

68. For the 'fatal claims' entitlement, SmartCore developed a process flow diagram (Figure 17), code, and an interactive dashboard (Figure 18), enabling hypothetical test cases to be run.

69. The tool was used by SmartCore at the same time as concept models, decision models, and rules statements were developed. At times, information from SmartCore was used to correct other artefacts.

Figure 13: rules statements (Oracle Policy Automation)

the claimant is eligible for weekly compensation for loss of earnings if

- the claimant is incapacitated because of a personal injury
- and
- any
  - the claimant is an earner at date of incapacity if
    - the claimant is an earner at the specific date
  - or
  - all
    - the claimant has incapacity during a policy period
    - and
    - the date of inquiry is on or after the date the claimant's application is made
    - and
    - the date on which the claimant's incapacity commenced is equal to or less than 5 years from a specific date
  - or
  - all
    - the claimant ceased to be in employment prior to date of incapacity
    - and
    - the claimant was incapacitated within 28 days from date ceased employment
    - and
    - any
      - the claimant is incapacitated within paid leave entitlement period following ceasing employment
      - or
      - arrangement has been made before the date of incapacity for the claimant to be employed within 3 months of ceasing employment
      - or
      - all
        - the claimant has employment within 12 months within ceasing employment as a seasonal employee
        - and
        - the claimant was employed by same employer in the two seasons before incapacity
        - and
        - the employer confirms it is likely they would employ the claimant in the next season
- and
- either
  - the claimant is not eligible for LOPE
  - or
  - all
    - the claimant is eligible for LOPE
    - and
    - the amount of LOPE payable to claimant is less than the amount of LOE payable

Figure 14: data interface (Oracle Policy Automation)

Text	Name	Value
Input Attributes:		
The claimant's date of ending employment is 4/04/2019.		2019-04-04
The claimant's date of incapacity is 13/02/2019.		2019-02-13
The claimant's date of starting employment is 1/01/2019.		2019-01-01
The date of inquiry is on or after the date the claimant's application is made.	Claimant_DateOfInquiryOnOrAfterApplicationMade	True
The claimant is incapacitated because of a personal injury.	Claimant_IsIncapacitatedBecauseOfPersonalInjury	True
The claimant has incapacity during a policy period.	Claimant_HasIncapacityDuringPolicyPeriod	True
The date on which the claimant's incapacity did not commence is equal to or less than 5 y...	Claimant_DateOnWhichIncapacityCommencedEqualToOrLessThan5YearsFr...	False
The claimant is not receiving payments under the organ donor act on the specific date.	Claimant_ReceivingPaymentsUnderOrganDonorActOnSpecificDate	False
The claimant is not on unpaid parental leave on the specific date.	Claimant_OnUnpaidParentalLeaveOnSpecificDate	False
The claimant is not eligible for LOPE.	Claimant_EligibleForLOPE	False
The claimant did not cease to be in employment prior to date of incapacity.	Claimant_CeasedToBeInEmploymentPriorToDateOfIncapacity	False
Was the claimant incapacitated within 28 days from date ceased employment?	Claimant_IncapacitatedWithin28DaysFromDateCeasedEmployment	<Unknown>
Was the claimant employed by same employer in the two seasons before incapacity?	Claimant_EmployedBySameEmployerInTwoSeasonsBeforeIncapacity	<Unknown>
Is the claimant incapacitated within paid leave entitlement period following ceasing employ...	Claimant_IncapacitatedWithinPaidLeaveEntitlementPeriodFollowingCeasingE...	<Unknown>
Is the amount of LOPE payable to claimant less than the amount of LOE payable?	Claimant_AmountLOPEPayableToClaimantLessThanAmountLOEPayable	<Unknown>
Has arrangement been made before the date of incapacity for the claimant to be employed...	Claimant_ArrangementMadeBeforeDateOfIncapacityForClaimantToBeEmploy...	<Unknown>
Does the employer confirm it is likely they would employ the claimant in the next season?	Claimant_EmployerConfirmsLikelyWouldEmployClaimantInNextSeason	<Unknown>
Does the claimant have employment within 12 months within ceasing employment as a sea...	Claimant_HasEmploymentWithin12MonthsWithinCeasingEmploymentAsSeas...	<Unknown>
Inferable Attributes:		
The claimant is in work for pecuniary gain on the specific date.	Claimant_InWorkForPecuniaryGainOnSpecificDate	True
The claimant is eligible for weekly compensation for loss of earnings.	Claimant_EligibleWeeklyCompensationForLossOfEarnings	True
The claimant is an earner at the specific date.	Claimant_IsEarnerAtSpecificDate	True
The claimant is an earner at date of incapacity.	Claimant_IsEarnerAtDateOfIncapcity	True
The claimant has a date of starting employment.		True
The claimant has a date of ending employment.		True

Figure 15: interview screen interface (Oracle Policy Automation)

The claimant is eligible for weekly compensation for loss of earnings. HIDE DETAILS

The claimant is incapacitated because of a personal injury.

The claimant is an earner at date of incapacity.

- The claimant is an earner at the specific date.
  - The claimant is in work for pecuniary gain on the specific date.
    - The claimant's date of incapacity is 13/02/2019.
    - The claimant's date of starting employment is 1/01/2019.
    - The claimant's date of ending employment is 4/04/2019.

The claimant is not eligible for LOPE.

Figure 16: test cases interface (Oracle Policy Automation)

Test Case	the claimant's date of incapacity	the claimant's date of starting unpaid parental leave	the claimant's date of ending unpaid parental leave	the claimant's date of starting receiving payments under the organ donor act
1	8/02/2019			
2	8/05/2019			
3				
4	8/02/2019	1/01/2019	1/04/2019	
5	8/05/2019	1/01/2019	1/04/2019	
6		1/01/2019	1/04/2019	
7	8/02/2019			1/01/2019
8	8/05/2019			1/01/2019
9				1/01/2019

Figure 17: process flow diagram (SmartCore)

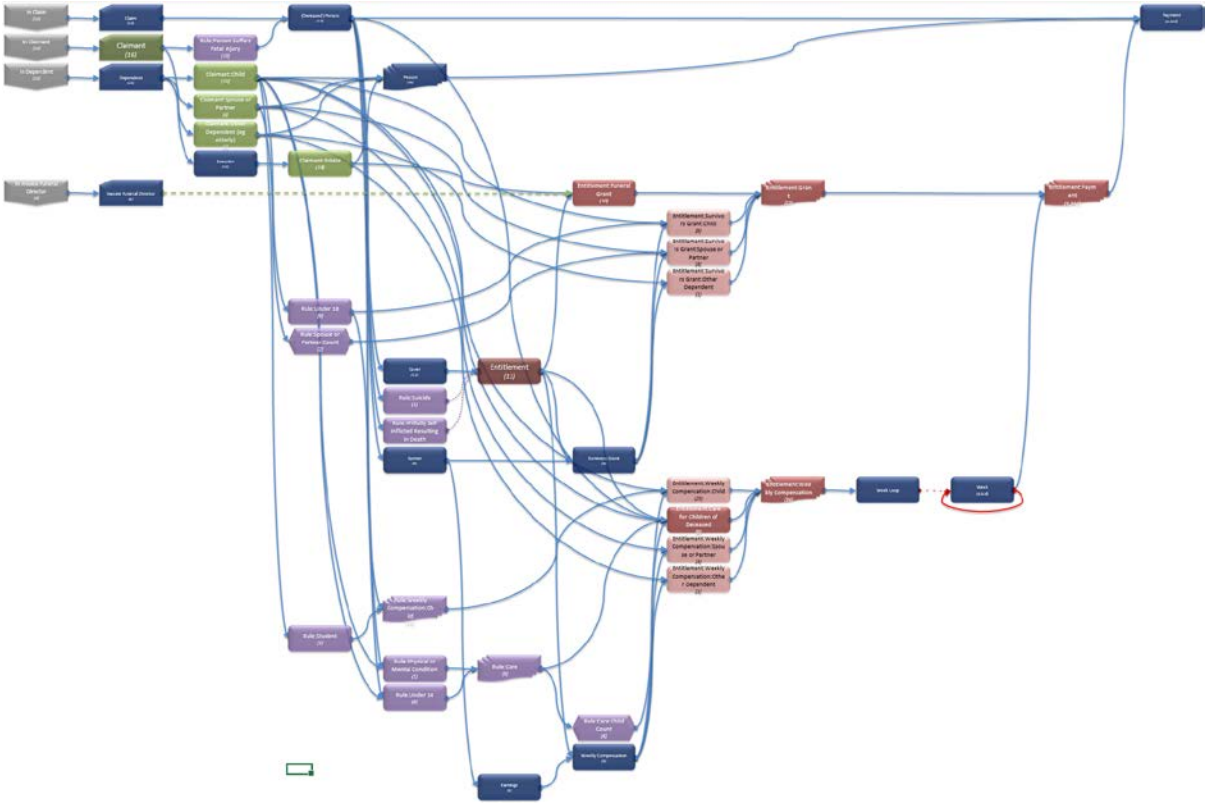
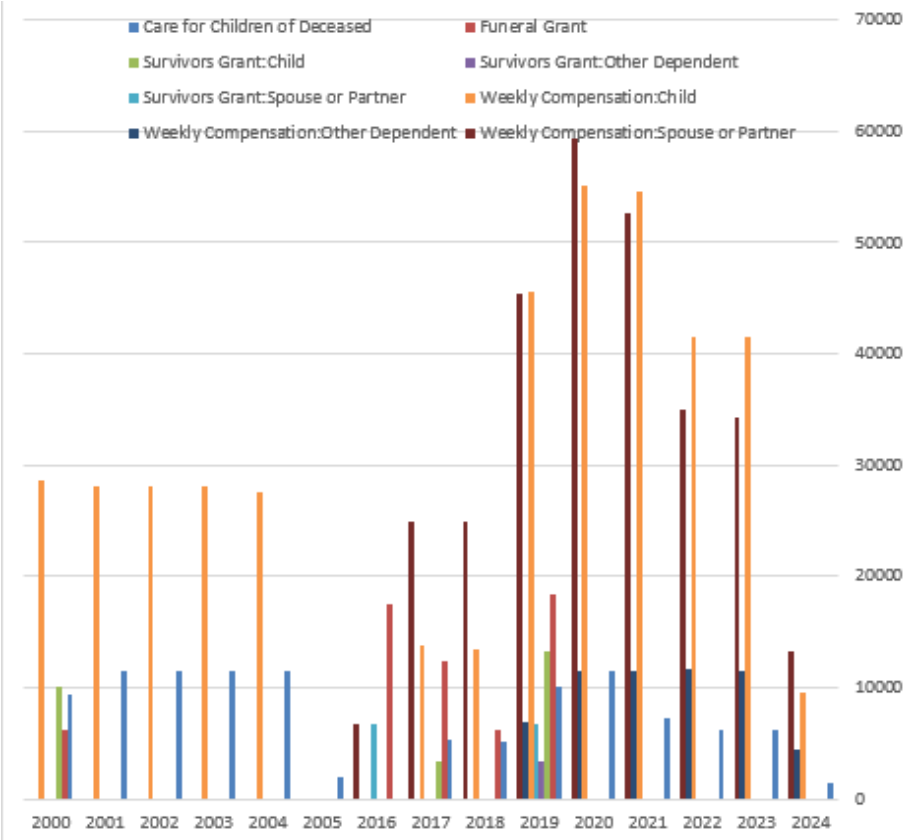


Figure 18: hypothetical payments over time (SmartCore)



7 Lessons

7.1 Lessons about the ways of working

Lesson 1. Writing code after rules statements not efficient

70. Our initial approach, of developing rules statements and passing those to software developers to write code, was inefficient. The software developers often had questions as they missed out on gaining a shared understanding from developing concept models and working side by side with the rest of the team. Also, the team missed an opportunity for the software developers to test the logic of rules statements through code as the rules are being created.

71. Developing the rules statements and code simultaneously, around the same table, improved the quality of the code and rules statements. And, while the team thought this approach might be more time intensive, the time taken to produce both outputs reduced.

Lesson 2. A need to consider who else is missing

72. What **Lesson 1.** revealed, beyond the need to code at the same time, is that the discovery had key people or disciplines missing from the collaboration. This highlighted that it is important to consider who else needs to be involved to achieve the desired outcomes, without introducing gaps between policy intent and what is implemented. For example, the discovery used operational subject matter experts to validate the rules statements, but it may have been preferable to use those experts collaboratively.

Lesson 3. It is necessary to use subject matter experts to validate rules statements

73. It became apparent that validating the rules statements was a necessary step. This is where additional subject matter experts were engaged to go through the rules statements, line by line. This process identified errors that the team had not previously identified, and often resulted in the subject matter experts learning something new.

74. Validation was not needed or used as a formal step for the concept models and decision models.

Lesson 4. It is important to not over-produce outputs

75. While acknowledging the value of storing artefacts digitally, there is value in recognising when the artefact is already good enough for the purposes needed and moving on to other tasks. Clear acceptance criteria and a mechanism to update criteria during a discovery sprint were found to foster continuous improvement.

76. This ensured the team maintained pace and removed unnecessary work from team members. From the outset, the team agreed on the ‘acceptance criteria’ of the outputs. This was simply a clear agreement on the required level of the outputs, considering the workload and availability of the team members during the experiment.

Lesson 5. There is a need to ensure outputs continue to be consumed and maintained

77. A discovery is an effective way to learn and produce value to many stakeholders. To realise this potential value, the discovery needed to ensure that the value is understandable to those not participating in the discovery and, where possible, built on other work.

78. To avoid the risk that this value is not realised, the team:

    a. held sprint review sessions

    b. consulted widely for existing material and included subject matter experts

    c. digitised and validated outputs with others outside of the core team

    d. held a showcase, which included multiple interactive information stations

    e. ensured summary documents were stored in a central location

    f. had clear product ownership.

Lesson 6. A small multidisciplinary team can produce most outputs

79. Most outputs from the discovery that would support a structural rewrite were developed with a small multidisciplinary team (eg, fewer than six people) with expertise in operations, policy, legal, business rules, and software development.

80. The discovery was fortunate to have many people involved who all added value, but the discovery worked to balance having a small, focussed and productive team against adding other necessary perspectives and subject matter expertise.

Lesson 7. The approaches used allowed the team to learn fast

81. The team operated by setting a defined time to carry out an activity (timebox activity). This was useful in ensuring team focus to reach the agreed output. This led to good productivity as a lot more activities could be completed.

Lesson 8. Value of different perspectives when looking at the logic of rules

82. The multidisciplinary team had subject matter experts in operational, policy, legal, and software development specialities. There was significant value from each of the different perspectives. The value of the perspectives was symbiotic in that it was not simply a case of subject matter experts sharing their knowledge – different specialities learnt from the timely sharing of perspectives.

Lesson 9. Full days or full-time commitment may have been less disruptive

83. The team agreed to meet for 20 hours per week (made up of five mornings) for six weeks. This was a pragmatic approach recognising that team members could not commit to 40 hours per week for three weeks with only a month's notice. This turned into fewer than 15 hours per week as the team did not meet every day, and often team members had to start late or finish early.

84. In week five (only), the team tried working two full days (16 hours) in lieu of five half days. This was less disruptive and appeared to be more efficient. It may have been more productive to delay the start of the discovery if that meant that team members could commit to either three weeks' full time or six weeks of two days per week.

Lesson 10. The scrum framework was helpful, but need to balance time servicing it

85. Using the rigour of the scrum framework supported the multidisciplinary team to deliver value, continuously improve, and have a common understanding of how team members could contribute.

86. However, there is a need to balance the time servicing its overheads. The discovery had fewer than 15 hours per week of collaboration time, and found that servicing the scrum framework was using up too much time. To balance this, the team agreed to stop daily stand-ups, stop sprint planning sessions, and have fewer demos.

87. The trade-off was that it was sometimes unclear what the team was working on, how long was available to complete that task, and what that item's comparative priority was. To manage this, the team increased the amount of information captured on Trello (the online project hub), and team members were asked to constantly document any progress to improve visibility of the broader project.

Lesson 11. A need to consider open source code

88. It would have been good to consider in more detail, before the discovery started, whether the discovery could publish its code on an open source platform to support the experiment (ie, publish code on OpenFisca, which is a common open source platform used internationally for these types of experiments). While OpenFisca was eventually used for some code, the risk profile was initially unclear to ACC, and it would have been more efficient to resolve this pre-discovery.

Lesson 12. A lack of cloud-based toolsets limited collaboration

89. Due to information security concerns, the team did not use cloud-based platforms for collaborative work on documents. This was a barrier in some instances, and use of these platforms would have enabled more efficient collaboration. It would have been good to explore this earlier, pre-discovery, and agree on a suitable platform. It would be good to explore the risk profile of using cloud-based tools and what platforms other agencies already use.

7.2 Lessons for a possible structural rewrite

Lesson 13. Legal drafters see significant value in rules statements

90. The rules statements allow the Parliamentary Counsel Office (legal drafters) to undertake a more comprehensive rewrite than if the rewrite was simply based on existing published law.

91. From a legal drafter's viewpoint, the benefits of rules statements for a rewrite are:

    a. the drafter needs to know what the law is now, and rules statements are a way of articulating this clearly and logically

    b. the analytical process involved in writing rules statements gives confidence that they accurately state the current law, and that can be reflected in drafting instructions

    c. gaps, inconsistencies, and ambiguities are exposed

    d. it helps to resolve what the law should be for those who will be issuing drafting instructions to fix issues

    e. the range of expertise is invaluable for getting robust results.

Lesson 14. Can now scope effort to blueprint other provisions

92. Based on the work during the discovery, those involved in a structural rewrite would be able to scope the time and resources needed to develop concept models, decision models, rules statements, and code for other areas of the AC Act.

93. Should the team need to manage or limit the areas explored, the team will be able to prioritise and sequence the remaining areas of the AC Act by balancing the areas that may benefit most from this work and those areas that are easiest to achieve.

Lesson 15. A lot of law is written from the perspective of ACC

94. While working on a decision model, the team were struggling to capture the logic. In the end, the team realised that this was complicated by the team's own subconscious decision to orientate the entitlement from the perspective of the claimant. This was in direct contrast to the way the AC Act was written, which (in this case) was from the perspective of ACC (a claimant's entitlement vs ACC's liability).

95. Witnessing this conflict in perspective raised the question as to why this area of the AC Act was written this way and opened the opportunity for future exploration with drafters as to which perspective makes more sense. It also highlights the need to review the team's unconsciously reorientating rules to a particular lens, as it risks misrepresenting the legislation.

Lesson 16. Legislation rules could get longer with more of them

96. A prevailing view had been that a structural rewrite would result in fewer or shorter rules to make the AC Act more accessible and human consumable.

97. However, and against this conventional view, ensuring there are no gaps in the rule logic may result in longer rules. This is expected to provide greater clarity and ensure legislation is more human consumable, and legislation in a machine consumable format could lead to more accessible law.

98. Furthermore, as legislation becomes machine consumable, and policy developers have tools to better understand its (positive and negative) effects, additional rules may be created to ensure more nuanced policy.

7.3 Lessons about outputs from the discovery

Lesson 17. Concept and decision models are valuable

99. Concept models enabled the team to quickly and efficiently understand the area they were looking at. It also brought the level of understanding within the team up to a shared baseline level, generating better discussions and contribution from all members.

100. Creating a concept model was also important to highlight the overall concepts and separate the thinking in the group from the process-focus that is often associated with legislation. This was important to develop a solid and flexible model that could successfully be used and referred to by the team throughout the discovery.

101. The approach used to create concept models (post-it notes on a whiteboard) made the process efficient and easy to pick up by all team members. At one point, the team experimented with not starting from a concept model, and quickly found that a necessary shared-understanding element was missing. This led to key areas of the entitlement not being covered.

102. An example of the impact of concept and decision modelling from the ‘fatal claims’ entitlement experiment was that the term ‘child’ was used in a manner that deviates from everyday use and is convoluted. Even in the limited context of the ‘fatal claims’ entitlement, the term is used in at least four different ways. Not using the exact words of the legislation, a child is defined as a person, in relation to their parent, with any of the following additional characteristics:

    a. The person is under the age of 14.

    b. The person is of any age if they have a mental or physical condition.

    c. The person is under the age of 21 if the person is studying.

    d. The person is under the age of 18.

103. Such varied contexts and definitions made the legislation more complex. This made the exercise more challenging, and made connecting related concepts more difficult. The broad overview and in-depth understanding gained from the concept and decision modelling made us aware of this issue and the team could confront this in its exercise and work with the current legislation. It also clearly illustrated opportunities for a structural rewrite.

104. While there is an opportunity to ensure legislation is more transparent, gaps in clarity are already managed through operational policy and practice.

Lesson 18. A need to keep assets

105. A challenge faced by the team was the on-going storage of concept and decision models. These assets are stored as digital images of the whiteboard sessions or as Visio diagrams, making them static and difficult to access. Not everyone has access to Visio, and the digital images can be difficult to read.

Lesson 19. Developing rules statements identified gaps

106. In the process of developing rules statements, the team identified gaps in the logic of the legislation that had not been previously identified or were not widely known.

107. An example from the ‘loss of potential earnings’ entitlement is that the AC Act refers to ‘full-time study or training’, but does not provide a definition for ‘training’ or full-time training’. Further policy work is required to understand whether a definition is needed.

108. While there is an opportunity to ensure legislation is more transparent, gaps in clarity are already managed through operational policy and practice.

7.4 Lessons about code

Lesson 20. There is an opportunity to code the whole AC Act

109. There is an opportunity to use a structural rewrite to articulate the entire AC Act in code and publish this on an open-source platform. This is to realise some of the potential of machine consumable legislation and is consistent with the work of the Better Rules initiative and Service Innovation Lab in New Zealand, which joins a growing push for this approach internationally.

110. This would take a larger step towards having machine consumable legislation in New Zealand, by having a substantive piece of legislation in code, which could be used as an example or template to follow.

111. This could be achieved by leveraging as a starting point the substantial body of code that may be produced as part of a structural rewrite. If MBIE, Parliamentary Counsel Office, and ACC apply the methods used in the discovery to other provisions of the AC Act to support a structural rewrite, a significant proportion of the AC Act would already be described in code to test and validate the logic of rules statements.

112. An articulation of the AC Act in code could then be available as a reference for agencies and, preferably, to fully realise the potential of machine consumable legislation, for the wider community.

113. This could be used for any number of initiatives, such as:

    a. supporting accident compensation policy development, by using operational data to model the effects and unintended consequences of a proposed policy change

    b. allowing the government to see how the AC Act interacts with other pieces of legislation (when they are also coded), to see system-level effects on, for example, tax and social welfare systems

    c. providing source code for consumers or advocates to develop an interface that answers the questions important to them

    d. allowing providers to integrate AC Act requirements into their systems or processes that engage with multiple legislative regimes.

Lesson 21. Code is an important part of validating rules statements

114. Like the gaps in the logic of legislation identified by developing rules statements (**Lesson 19.**), the process of coding the rules statements also identified gaps in logic that had not been identified. That is, having a machine test the logic based on code checks the accuracy of the rules statements.

115. Examples from the ‘social rehabilitation’ entitlement identified by code:

    a. Avoid double-negatives in drafting (they are difficult to comprehend).

    b. For child care, add an explicit requirement for the claimant to have children.

    c. Possibly more clearly define ‘any other child’ in clause 12, schedule 1.

    d. Section 67 (‘Who is entitled to entitlements’) seems redundant and not aligned with section 48 (‘Person to lodge claim for cover and entitlement’).

    e. Define “claimant’s children” in clause 15(1)(b), schedule 1.

Lesson 22. Value of code versus coder

116. The software developer brings a lot of value to the multi-disciplinary team, in addition to the code they create. They bring a different perspective or way to look at the logic that increases the robustness of the rules statements.

Lesson 23. The machine has an absence of understanding

117. The value in having to explain the policy to a machine is that humans can leverage the machine's complete lack of assumption and context to bring to light the gaps in the policy that the human brain has overlooked.

Lesson 24. The interface demonstrated the potential for policy tools

118. The interface (or dashboard) developed for the 'loss of potential earnings' entitlement demonstrated how machine consumable legislation has the potential to provide a tool to support policy development.

119. The interface allowed the user to change the date of entitlement and see the effects that change had on how many were eligible, the types of injuries they had, and the severity of those injuries. The experiment used example data, but could be easily swapped out for real world data.

120. The example was elementary, but was created to illustrate a future where policy development could use rule sets (legislation as code) to help in researching new opportunities or refinements to both existing or new legislation and its impact.

Lesson 25. It is easy to work with test cases

121. Test cases were first created with decision models and were later used as a blueprint for test cases of the rules statements and code validation. It was easy to work with these test cases to validate the team's work as it progressed and to build on this suite of tests as the discovery unpacked more of the entitlements in the AC Act.

Lesson 26. It is useful to identify if dates are static or change

122. It should be made clear at the outset of the coding process whether an entitlement will vary depending on the date it is being considered. This will ensure the coding process is more efficient, as coding requires time bound definitions.

Lesson 27. Decisions requiring human judgement can be shown in code

123. It is often perceived that decisions requiring human judgement within existing legislation are difficult for software developers to code, so the team decided to tackle a discrete piece of legislation to test this.

124. The team found that allowing the logic models to defer to human input was as simple as creating concepts that represented human input. Furthermore, this becomes a way to track the quantity of human judgement elements within a piece of legislation, and the limits and constraints under which this judgement must operate. This could provide a new perspective and way to measure legal instruments by the amount of human judgement they introduce.

Lesson 28. There need to be agreed standards and architecture

125. Code developed for the discovery using software developers was made publicly available on an open-source platform (in this case, OpenFisca). While this was good as part of the experiment, the platform does not provide authoritative code that could be relied upon and the code would unlikely be updated as it has no responsible owner. Furthermore, without agreed standards and architecture, other experiments may use different languages and platforms, meaning New Zealand is not taking steps towards having a single repository of code.

126. Code was also developed using Oracle Policy Automation, which does not have web API (application programming interface) functionality. This limits the ability for the government to view multiple pieces of legislation as a single system. It also means that the code can only be used by the agency with the licence, and does not allow the public to use the code, which should be key requirement for machine consumable legislation given legislation is traditionally open in a democracy.

127. If, each time code was developed, it used agreed standards and architecture, then New Zealand would take a small step towards having a single repository of reliable code on an open source platform for all its legislation. At the time of writing this report, the Better Rules initiative had standards and architecture on its annual work plan, which should resolve this.

7.5 Lessons about the tools used

Lesson 29. Different types of tools are required

128. There were several concepts regarding tools that came into focus during the discovery:

    a. A need for capturing, storing, and referencing the concept and decision models and their relationship to the legislation and business rules to better preserve institutional knowledge.

    b. A need for test suite tooling that can be used in policy development through to operations to assist in the development of a robust approach to policy development and delivering on policy intent.

    c. Understanding the distinction between published legislation as code ('1:1 legislative match') and internal government agency business rules systems and the implications for the tools that will be publishing them.

    d. The question of storing rules in a dataset form from which executable legislation as code could be derived.

Lesson 30. Experimenting with Oracle Policy Automation

129. Oracle Policy Automation demonstrated how a tool can be used to quickly turn rules statements into natural language code (referred to in the Better Rules Discovery Report as 'pseudo-code') that can be swiftly turned into a web interface or integrated with the models in a tool. Oracle Policy Automation was also good for developing test cases that can be easily understood by subject matter experts or run in other applications.

Lesson 31. Experimenting with SmartCore

130. SmartCore demonstrated how a single tool can capture, manage, and manipulate the concept models, decision models, rules statements, and code. In the experiment on the 'fatal claims' entitlement, the line of sight between concept model and code was maintained.

Lesson 32. Tools can be complementary

131. When the discovery used different tools, the goal was to explore the value of each of them, and it was not expected that the tools would support each other. But, for the 'loss of potential earnings' entitlement', test cases developed by Oracle Policy Automation were mailed to and consumed by SmartCore. This suggests that there are overlapping roles of tools that could be explored to find even more efficient means of developing code, models, interfaces, etc.

Lesson 33. Tools can develop code almost simultaneously

132. The tools proved there can be low latency between development of the concept model and the first iteration of code using these tools. It is almost instantaneous.

Lesson 34. There are limitations of tools used

133. SmartCore produces SQL (structured query language) code from a spreadsheet, which needs to be inputted into something to be used.

134. Equally, Oracle Policy Automation is a reasonable stable license tool that requires quite a lot of learning, is expensive, and does not have web API (application programming interface) functionality.

135. OpenFisca is a rules execution machine for publishing legislation on the internet. It is not designed as a business rules engine in a large organisation.

Lesson 35. Value of visualisation tools

136. Visualisation tools, such as Drawing Express Lite, are free and easy to use tools that can be used to digitise concept models and other diagrams while they are being created. This saved later write up.

Lesson 36. Tools highlight that developers are more than just code writers

137. Mentioned in **Lesson 22.**, developing code using tools and without software developers highlighted some of the value that software developers bring. They look at rules and logic on behalf of the machine and improve the logic of rules statement.

Lesson 37. Visio is good for standardisation, but has limitations

138. Some of the concept and decision models were documented as Visio diagrams. Within Visio models can be produced to an agreed standard, using a range of templates and stencils.

139. However, the challenges included:

    a. restrictions on access: with Visio being a licensed product, not all team members had access to the tool, which limited their ability to view and contribute

    b. different Visio versions: different team members had access to different versions of Visio, which made sharing artefacts difficult, as the team needed to save them according to the version of the team member

    c. knowledge and application of the standards: with the artefacts being of a defined standard, the lack of this knowledge of the standards limited the ability for others to create the artefacts – this created a bottleneck on the production of the artefacts, and the team did not fully explore the richness of the modelling, due to limited time in understanding the standards.

Lesson 38. Potential to use machine learning to map existing legislation

140. A piece of work which will be ongoing is exploring the representation of existing legislative text in a visual format that highlights its structure and references to provide insights on larger Acts facing a restructure. Work at this stage is still very experimental and there is a lot of discussion happening at an international level as to what may be the best approach towards achieving this.

Lesson 39. Published law and business rules have different attributes

141. There was discussion between the members of the team, specifically the business rules experts, the legal teams, and the software developers who have been working on legislation as code. Out of that discussion a number of interesting threads to follow up on came to light.

142. The discussion was centred around the use of rules in software and the different requirements depending on the focus of the code being written. An example used in the discussions was the rule that ACC had to respond to a request within 21 days. The business rules experts however had to code that number as 15 days as ACC internally had decided that that would be the most suitable number for operations.

143. This layering of logic for internal operations is like the effect case law has on legislation in introducing new boundaries or approaches to how the law is interpreted. It would be normal to assume the ruleset that makes up the business rules would be much larger than what would be required to describe the legislation in code.

144. The idea that government agencies could utilise the published 1:1 legislation as code as a foundational layer for their business rules needs further investigation. This would introduce a potentially new requirement on published legislation as code; technically it should allow third parties to build rulesets that expand on it.

Lesson 40. There is a potential in name spacing

145. It was common while the group was coding to find the AC Act's structure is at times a little bewildering. Within software this is often solved using a concept known as 'name spacing'. Name spacing is using a hierarchal tree-like structure of terms that indicate where a particular piece of code is in the structure. This approach allows for multiple definitions of the same term, as currently happens.

146. The current equivalent solution for this approach is the numbering / lettering system. Although this offers very little assistance in the use of terms in use elsewhere, it does provide clear context of the structure of the document.

147. For example, within Inland Revenue's legislation there is a section titled "Tax credits paid in cash". Under that section is another titled "Abating WFF tax credits" and then under that section another titled "Best Start tax credit". To reference the start of the "Best Start" section currently would require the following: "MG.1". This is meaningless to someone new to the document. This current system could be preserved and name spacing introduced in parallel. Such a system could result in a reference that would look something like:

    Tax credits paid in cash > Abating WFF tax credits > Best Start (MG.1)

148. Ensuring this form of structure would be beneficial in helping retain control of the document's organisation during redrafts. It would also help readers of the document to understand the structure and where to find topics.

149. Finally, it would aid considerably in structuring a coherent and organised 1:1 presentation of the instrument in code. This is a topic for further exploration with the drafters.

8 For more information

The following is not exhaustive, but provides some further information.

Data61 | CSIRO (no date): *A world of opportunities where laws & regulation become digital* (https://digital-legislation.net/)

Digital.govt.nz (March 2018): *Better Rules for Government Discovery Report* (https://www.digital.govt.nz/showcase/better-rules-for-government-discovery-report/).

Digital.govt.nz (no date): *DISCUSS.DIGITAL.GOVT.NZ* (https://discuss.digital.govt.nz/dashboard)

Digital.govt.nz (3 May 2017): *What is the Service Innovation Lab? Come and see for yourself!* (https://www.digital.govt.nz/blog/what-is-the-service-innovation-lab-come-and-see-for-yourself/).

Digital.nsw (uploaded 10 April 2019): *Rules as Code showcase* (https://www.vimeo.com/329720314)

FutureLaw 2019 (uploaded 12 April 2019): *Government as a Platform* (https://www.youtube.com/watch?v=5EgO7WtMXYA)

Google Docs (14 May 2019). *Rules as code show & tell notes* (https://docs.google.com/document/d/1d0lHW87lgTjpHe88WAfBS3g0wJrPH62ySN2RUSmM0MQ/edit)

IT World (1 April 2015): *France's laws are now on GitHub* (https://www.itworld.com/article/2904079/france-s-laws-are-now-on-github.html).

Love, N, Genesereth, M R (Proceedings of the International Conference on Artificial Intelligence and Law, 2005): *Computational Law* (http://logic.stanford.edu/publications/love/computationallaw.pdf?utm_source=user_mailer&utm_medium=email&utm_campaign=catch_up)

MBIE (no date): *Better rules – Better outcomes* (https://www.mbie.govt.nz/business-and-employment/business/support-for-business/better-for-business/better-rules-better-outcomes/).

MBIE (no date): *Holidays ACT Review* (https://www.mbie.govt.nz/business-and-employment/employment-and-skills/employment-legislation-reviews/holidays-act-review/).

OECD Observatory for Public Sector Innovation, and the United Arab Emirates Mohammed Bin Rashid Centre for Government Innovation (February 2019): *Embracing Innovation in Government: Global Trends 2019* (https://trends.oecd-opsi.org)