Explainable AI (XAI) in Rules as Code (RaC): The DataLex approach

Andrew Mowbray, Philip Chung and Graham Greenleaf[*]
13 May 2022 – Pre-publication draft, comments are welcome to andrew@austlii.edu.au

Abstract

The need for explainability in implementations of 'Rules as Code (RaC)' has similarities to the concept of 'Explainable AI (XAI)'. Explainability is also necessary to avoid RaC being controlled or monopolised by governments and big business. We identify the following desirable features of 'explainability' relevant to RaC: Transparency (in various forms); Traceability; Availability; Sustainability; Links to legal sources; and Accountability. Where RaC applications are used to develop automated decision-making systems, some forms of explainability are increasingly likely to be required by law. We then assess how AustLII's DataLex environment implements 'explainability' when used to develop RaC: in open software and codebases; in development and maintenance methodologies; and in explanatory features when codebases are executed. All of these XAI aspects of DataLex's RaC are consistent with keeping legislation in the public domain no matter how it is encoded.

Contents

[*] Andrew Mowbray is Professor of Law & Information Technology, University of Technology Sydney and Co-Director, AustLII; Philip Chung is Associate Professor of Law, UNSW Sydney and Executive Director, AustLII; Graham Greenleaf is Professor of Law & Information Systems, UNSW Sydney, and Senior Researcher, AustLII.

The authors have received valuable comments from Matthew Waddington, but all responsibility for content remains with them.

1. Introduction

This article explores the fundamental need for explainability and transparency in implementations of 'Rules as Code (RaC)' and examines the similarities of this requirement to the concept of 'Explainable AI (XAI)'. We will argue that the intrinsic value of explainability associated with XAI is also essential for RaC to be fully effective. It is also necessary if RaC is to contribute to the effectiveness of the rule of law and transparent decision making and avoid being controlled or monopolised by governments and big business.

We will then introduce AustLII's DataLex environment and discuss how 'explainability' is implemented as part of this platform when used to develop RaC applications in a number of ways: in open software and codebases; in development and maintenance methodologies; and in explanatory features when codebases are executed. All of these XAI aspects of DataLex's RaC are consistent with keeping legislation in the public domain no matter how it is encoded.

2. Rules as Code (RaC), and who will control it?

'Rules as Code' can be described as[1] the activity of creating or transforming a legal text which is in a natural language (legislation, regulations, or other legal instruments – generically, 'law' or 'rules'), in or into a representation in a computer-processable form (code). RaC is also applicable to other types of law-related rules such as standards, as well as organisational rules including codes of practice, codes of conduct and business procedures. Applications include the embedding of rules in automated systems so as to facilitate the communication between systems (and possibly humans) to determine such matters as design and compliance. Another type of RaC applications allows a human user to interact with systems in relation to specific fact situations and thereby produce conclusions which are an accurate statement of the legislative intent of the legal text when applied to that scenario. RaC is also often referred to as the creation of machine-consumable ('executable') versions of laws and other type of rules, or 'creating machine-interpretable regulation'.[2] However, Waddington points out that, although RaC and those interested in it tend to have a practical orientation, RaC does not necessarily require a program that implements the legislation to achieve some purpose, and it will still be valuable as RaC even if it is only used as a representation of the legislation to help achieve legislative clarity and avoid drafting errors ('coding before enacting').[3]

Rules as Code shares many similarities with rule-based approaches in artificial intelligence ('symbolic AI'). This is distinct from the 'sub-symbolic AI' systems based on neural networks, statistical inferencing and machine learning (ML). These approaches do not usually involve the construction or manipulation of rules. AI has become almost synonymous with 'machine learning' in many people's minds, so it is important to stress at the outset that there is another form of AI, symbolic AI, that has little to do with ML. We will see later that 'explainable AI' (XAI) also has quite a different meaning when we are discussing symbolic AI (and RaC) rather than neural networks and ML.

Part of the history of RaC is that it had a close relationship to research into 'expert systems', which was dominant in AI in the 1980s.[4] One focus of expert systems research was on how to capture the expertise of specialists in various technical fields ('domain experts'), including areas of law where it was considered valuable to capture the knowledge of experienced practitioners in specialised fields.

---

[1] This definition is by the authors. Terminology in this field is still very unsettled, with no generally accepted definitions for 'Rules as Code', 'Law as Code' or 'Code as Law'.

[2] Productivity Commission (Australia) *Information Paper on Regulatory Technology* (October 2020), p. 13 <https://www.pc.gov.au/research/completed/regulatory-technology/regulatory-technology.pdf>

[3] M. Waddington 'Research Note: Rules as Code' *Law in Context* 3 January 2021, p. 180 <https://doi.org/10.26826/law-in-context.v37i1.134>

[4] Russell, S and Norvig, P, *Artificial Intelligence: a Modern Approach*, 4th Ed, Pearson 2021, Chapter 7

A different focus was to start with the legislation in a field, and attempt to convert it (and expert knowledge about it) comprehensively into an application which could answer many legislation-oriented questions. Logic programming, often involving the use of Prolog,[5] was a significant component of this legislative focus. By 2000, this wave of interest in automation of legislation seemed to have run its course but left many valuable lessons from two decades of experiment and scholarship.[6]

Since around 2015, there has been renewed interest and enthusiasm for the computerised representation and/or automation of legislation referred to as 'Rules as Code' (RaC).[7] Perhaps the most significant stimulus in relation to RaC is that various governments have started to develop apps based on legislation in their jurisdictions. Examples are projects or reports in New Zealand,[8] France,[9] New South Wales,[10] Jersey[11] and Canada,[12] and a report for the OECD.[13] Despite this interest, government involvement in RaC remains tentative as yet, somewhat reminiscent of the hesitant government adoption of digitisation and Internet publication of legislation in the mid-1990s. There are few running applications to demonstrate developments to the public.

This RaC momentum (it is premature to call it a 'movement') also comes from outside government, involving rapidly developing conferences and events to propagate interest[14], websites to survey available software, and instructional materials and events.[15]

There is an important distinction between the post-2000 resurgence of interest in the uses of artificial intelligence (AI) in all fields, and the later growth of interest in RaC. The general 'AI spring' has been to a large extent centred on enormous advances in the technology and uses of machine learning (ML) and neural networks of many types. However, in relation to law, advances in ML have had a much more limited effect. 'Rules as Code' applications are often based on methodologies similar to 'symbolic AI', described as consisting of 'symbols, combinations or symbols, and rules and

---

[5] M. Sergot, F. Sadri, R.A. Kowalski and F. Kriwaczek 'The British Nationality Act as a logic program' Communications of the ACMVolume 29 Issue 5 May 1986 pp 370–386 <https://doi.org/10.1145/5689.5920>

[6] G Greenleaf, A Mowbray, and P Chung 'Building sustainable free legal advisory systems: Experiences from the history of AI & law' *Computer Law & Security Review*, 34:314–326, 2018 <https://www.sciencedirect.com/science/article/abs/pii/S026736491830075X>. Pre-publication version at [2017] UNSWLRS 53 <https://papers.ssrn.com/abstract_id=3021452>.

[7] Other terminology used to describe part or all of the field include 'Law as Code', and 'computational law'.

[8] Government of New Zealand. *Better Rules for Government Discovery Report* March 2018 <https://www.digital.govt.nz/dmsdocument/95-better-rules-for-government-discovery-report/>; See also for project details, H. Fraser *What is Better Rules?* Government of New Zealand, December 2019 <https://www.digital.govt.nz/blog/what-is-better-rules/>

[9] See <https://openfisca.org/en/>.

[10] See <https://www.nsw.gov.au/media-releases/digitising-rules-of-government-to-make-compliance-easy>.

[11] Waddington, M, "Machine-consumable legislation: A legislative drafter's perspective – human v artificial intelligence" 2019 (2) *The Loophole - Commonwealth Association of Legislative Counsel* 21

[12] S. McNaughton 'Week 64 — The State of Rules as Code in the Government of Canada' blog post 30 May 2020. <https://medium.com/@mcnaughton.sa/week-64-the-state-of-rules-ascode-in-the-government-of-canada-8f3cb327448d>

[13] J. Mohun, J. and A. Roberts *Cracking the code: Rulemaking for humans and machines*" OECD Working Papers on Public Governance, No. 42, 2020, OECD Publishing, Paris https://doi.org/10.1787/3afe6ba5-en; See also Greenleaf, G., Mowbray, A. and Chung, P, 'Strengthening Development of Rules As Code: Submission to the OECD's OPSI on Cracking the Code' (June 23, 2020) <https://papers.ssrn.com/abstract_id=363877>.

[14] For example, Australasian Society for Computers and Law (AUSCL), *Rules as Code 2.0 - Global plenary for networking and co-designing solutions to RaC's and society's grand challenges*, 15 March 2022; AUSCL, *Rules as Code Masterclass 3: AustLII DataLex*,19 May 2021, <https://youtu.be/8fTqQcVyVa>.

[15] In addition to the OECD Working Paper mentioned above, see also Tom Barraclough, Hamish Fraser and Curtis Barnes, *Legislation as Code for New Zealand: Opportunities, Risks and Recommendations*: Tim de Sousa *Rules as Code Handbook,* 2021 <https://github.com/Rules-as-Code-League/RaC-Handbook/wiki/3-Building-trustworthy-systems-using-coded-rules>.

operations on symbols',[16] an approach that dominated the field of AI for three decades from the 1960s, until 'sub-symbolic' neural networks coupled with ML became dominant.

2.1. RaC's relationship to free access to law: Danger or opportunity?

Since the early 1990s, there has been nearly 30 years of gradual acceptance and adoption by governments around the world of 'free access to law'. This primarily means that governments have stopped exercising monopoly powers over dissemination of legislation and case law ('primary legal materials'), and have both themselves published these primary materials for free public access, and have also facilitated its republication by others, including by both commercial and non-commercial publishers.

The sixty members of the Free Access to Law Movement (FALM),[17] mainly non-profit academic bodies, and other types of NGOs, have played a leading role in this success,[18] both by becoming the largest publishers of free access law in many jurisdictions, and by helping convince governments (by both example and argument) that it was no longer socially acceptable to attempt to make money out of the dissemination of the laws that govern us.[19]

In addition to free access to primary legal materials, there has been a parallel 'open content' movement to create free access to academic and professional commentary on the law, with success varying a great deal between jurisdictions. For example, the Legal Scholarship Network (part of SSRN)[20] has preprints of over 330,000 academic legal articles; Lexology aggregates the professional commentary of hundreds of law firms every day;[21] AustLII provides the complete content of over 120 Australasian law journal (and other LIIs do so on a lesser scale); CanLII Connects provides daily commentary from law firm authors on Canadian law; and both legal textbooks and professional handbooks are increasingly available from LIIs.

In summary, this thirty year history has created free access to two main types of legal materials, namely primary legal materials and a significant body of expert commentary on law. The relevance of this history to RaC is that we can distinguish three types of digitisation of legal information relevant to the giving of professional quality legal advice: (i) representation of primary legal information used by legal advisers; (ii) representation of expertise and its general application (such as in textbooks and law journal articles); and (iii) representation of law[22] in such a way (eg code) that it can be applied to individual situations. These categories overlap in reality, but these distinctions enable us to consider more precisely how likely is it that each category will become part of a 'commons of legal expertise',[23] legal information (including RaC code-bases) that anyone can use in order to resolve

[16] M. Mitchell *Artificial Intelligence: A Guide for Thinking Humans* (Pelican, 2019) pp 11-12.

[17] Free Access to Law Movement (FALM) website <http://www.falm.info/>

[18] Graham Greenleaf, Philip Chung, and Andrew Mowbray 'Update: Legal information institutes and the free access to law movement' *Globalex*, February 2018 <http://www.nyulawglobal.org/globalex/Legal_Information_Institutes1.html >.

[19] Graham Greenleaf, Andrew Mowbray, and Philip Chung 'The meaning of 'free access to legal information': A twenty year evolution'. *Journal of Open Access to Law* (JOAL), 1:1, 2013.< https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2158868>.

[20] Legal Scholarship Network (LSN) < https://www.ssrn.com/index.cfm/en/lsn/>

[21] Lexology < https://www.lexology.com/>

[22] We omit those instances where expertise about law, particularly on interpretation of legislation, is also included in what can be applied to an individual situation. Including such expertise involves additional complications which are beyond the scope of this article. We are limiting 'Rules as code' to refer to statutory rules only. Systems which aim to include expertise in interpretation and the like are best referred to as 'expert systems'.

[23] For a more detailed argument, see Graham Greenleaf 'Review essay – technology and the professions: Utopian and dystopian futures' *UNSW Law Journal*, (2017) 40(1):302–321. <https://ssrn.com/abstract=2973244 >

legal problems.[24] This ideal has also been described as 'a publicly available community repository of legislation as code to build upon and test against'.[25]

Primary legal materials are now effectively in the public domain,[26] as is a significant percentage of legal expertise in textual form. The danger now is that the third stage of online legal information – representation of law in a form that can be specifically applied (including RaC) – will not become part of the public domain of free access legal materials. If Rules as Code become almost entirely the property of governments and/or large private sector organisations (law firms, consultancies and publishers) the future of free access to law will be one of failure. If this happens, then RaC will have been monopolised, in much the same way as online access to any legal information was monopolised before the mid-1990s.

The meaning of 'free access to law' is not static.[27] Its underlying moral justification, that individuals (and their advisers) should be readily and freely able to ascertain the laws that govern them, is as applicable to the presentation of a statute as RaC as it is to that same statute in textual form. The provision of justice in the 21st century requires that providers of free legal advice also have effective access to Rules as Code resources, and the ability to use and modify those resources to serve their own needs.

There will be many RaC applications which it will be valuable for individuals to access directly, such as systems which concern provision of government benefits, or calculate tax liabilities, or concern consumer disputes, but there will also be many RaC applications which will mainly be of assistance to those who provide free legal advice to individuals. In both cases these RaC applications will be, in effect, in the public domain. We have argued previously that free access legal information institutes (LIIs) should have a role to play in providing access to these systems to free legal advice providers, such as community legal centres and other legal aid providers.[28]

'Effective access' by these free legal advice providers will also require access to explanatory mechanisms that are congruous to 'explainable AI' in relation to law, as we will argue in the next section.

2.2. Government RaC: building blocks of law's public domain

The role of governments in expanding the availability of RaC in the public domain is crucial. Governments everywhere are increasingly likely to produce legislation codebases for their own purposes. These include for the work of government-run free legal advice providers, for provision of government benefits, and also for the internal work of many government agencies. These government-produced codebases should be the building blocks, within a particular jurisdiction, of the RaC public domain for that jurisdiction. For this to be effective requires two steps.

First, like traditional legislation, RaC codebases need to be published under licences that make them part of the public domain. Such 'commons licences' can take various forms. For example, Australia's federal government publishes textual works produced by government (reports, discussion papers etc) under a Creative Commons BY 4.0 licence,[29] which allows them to be re-published, modified, and

[24] Greenleaf, ibid, Part IV 'Conclusions: Can we create a "commons" of expertise?'.

[25] Pia Andrews 'Exploring and advising public sector reform across the ANZO region' *Pia Andrews blog*, 21 March 2022 <http://pipka.org>.

[26] Greenleaf G and Lindsay D. *Public Rights: Copyright's Public Domains* (Cambridge University Press, 2018).

[27] Greenleaf G, Mowbray A, and Chung P. The meaning of 'free access to legal information': A twenty year evolution. Journal of Open Access to Law (JOAL), 1:1, 2013. <https://papers.ssrn.com/abstract_id=2158868>.

[28] Mowbray A, Chung P and Greenleaf G, 2020, 'Utilising AI in the legal assistance sector—Testing a role for legal information institutes', *Computer Law & Security Review*, vol. 38, pp. 1 - 9, <http://dx.doi.org/10.1016/j.clsr.2020.105407>.

[29] Creative Commons 'Attribution 4.0 International (CC BY 4.0)' < https://creativecommons.org/licenses/by/4.0/>

even used for commercial purposes, provided the government is attributed as the copyright owner.[30] By making RaC codebases re-usable by others, governments can contribute essential 'building blocks' to the 'commons of legal expertise' that we advocate.

Second, RaC codebases produced by government should aim, as far as possible, to represent code generically so that it can be used in a variety of ways to suit the implementation methods of different applications and for a range of purposes. This will ensure that the code of the legislation is separate from the code of its implementation (for example, in provision of government services). This separation will allow other uses of the legislation code to incorporate it into their own codebases without having to first disentangle it from government implementation code.

Waddington, deputy head of Jersey's Legislative Drafting Office, with a strong interest in RaC, stresses this second requirement:

> The "coding" that the government publishes would not be an executable program that implements the law automatically in some way. Instead others would be producing executable programs for their own particular purposes, … That means the version published by government would need to be program-neutral as far as possible. … The point is that Rules as Code is not wedded to any of these potential technology solutions so much as to the idea that the "coding" (or mark-up) of the legislation should be widely usable, traceable to the legislation, rather than adding material to reflect assumptions about procedures or implementation. [31]

Waddington then takes this argument further:

> As a legislative drafter, I am on the wing of the Rules as Code movement that aims, as far as practicable, not even to include material derived from other legislation (as opposed to coding that links to the coding of another statute, where there is a cross-reference in the text or where an Interpretation Act applies) or from common law or other sources.[32]

As he says, just because other sources of law are necessary for full understanding, does not stop the publication of items of legislation without all the sources of interpretation necessary to understand them. The same can apply to encoded legislation (RaC in Waddington's view). If governments then want to use the encoded legislation to provide some online service, then at that point they can add procedural code that focuses on the key aspects of the RaC component needed to provide the service, and other materials such as case law which enable to the RaC to be properly interpreted.

3. Explainable AI (XAI): To whom?

3.1. The demand for 'explainability' in AI

'The vast majority of AI systems' it is claimed 'have a degree of opacity that makes it hard to understand how the algorithmic decisions or predictions of the system have been reached'.[33] This gives rise to the notion of 'black box AI', referring to 'scenarios in which we can see only the input data and the output data … without having insight into exactly what happens in between.'[34] This is particularly so for AI systems using neural networks (of many varieties) where system developers have little idea what happens in the 'hidden layers' of the network. Machine learning (ML) is used to build these hidden layers, and most AI built using ML will have this 'black box' problem. Systems

[30] Greenleaf and Lindsay op cit pp. 493-498.

[31] M. Waddington 'Research Note: Rules as Code' *Law in Context* 3 January 2021, pp 181-182 <https://doi.org/10.26826/law-in-context.v37i1.134>

[32] Waddington, op cit, p. 183.

[33] Martin Ebers 'Regulating Explainable AI in the European Union. An Overview of the Current Legal Framework(s)' in: L. Colonna and S. Greenstein (eds.), *Nordic Yearbook of Law and Informatics 2020: Law in the Era of Artificial Intelligence*, p.4 <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3901732>.

[34] Ibid, p.3

based on symbolic AI will not have the 'black box' problem, but its absence will not necessarily mean that they are explainable.

We require 'explainability' of computing systems that have important consequence for our lives for a number of reasons, but those reasons are highly dependent on our relationship to the system:[35]

- Those individuals affected directly by a decision or prediction will want to understand how it was reached, in order to be convinced of its fairness (among other things), and if they are using the system directly, the explanation will need to be understandable by them;
- Organisations utilising such systems need to understand how decisions they administer are determined, and (usually) to keep a record of this, with the explanation being understandable to those who run the system, and also to any parties who review such decisions;
- Such organisational users of AI systems also need to encourage individual users of the systems they provide to trust (where justified) the outcomes of use of those systems, and this trust may depend on explainability;
- System designers (and those who have to maintain systems) need to understand how systems work in order to improve them, or to debug them where necessary;
- All parties involved will need to be convinced that the system is operating in ways which do not breach legal requirements (see below);
- 'Social licence' for such AI to be used in making important decisions requires a level of trust from society as a whole, which is unlikely to occur with a 'black box'.

Most AI research on explainability focuses on ML, and definitions of explainability often assume ML.[36] However, explainability is just as important for symbolic reasoning systems although different explanatory mechanisms will be needed. The same sort of mechanisms are also likely to be important for RaC based applications.

3.2. Legal requirements for explainability

In many countries, some form of 'explainability' has now moved beyond being desirable to being legally required. What we will see is that many Rules as Code applications, because they deal with the application of legislation to human 'data subjects' are very likely to come within the scope of these legislative requirements.

The best known such requirement is article 22 of the EU's *General Data Protection Regulation* (GDPR) of 2016,[37] although it does not directly require explainability but instead imposes other controls on automated decision-making (ADM), with direct effect on 30 countries. Article 22(1) provides that 'The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her.' After making exceptions for where such decisions can be made, Article 22 then provides that in such cases there must be 'suitable measures to safeguard the data subject's rights and freedoms and legitimate interests, at least the right to obtain human intervention on the part of the controller, to express his or her point of view and to contest the decision' (art. 22(3)). Further restrictions are then placed on where 'special categories of personal data' (sensitive data) are proposed to be used (art. 22(4)).

Without going into any detailed analysis of Article 22, it is obvious that 'suitable measures' to protect data subjects could include explainability of decision, and that it would be difficult to 'contest the decision' if it is not explainable. GDPR article 22 (and articles 13-15) is an imperfect way of requiring

[35] Ibid, p.4

[36] See for example Wikipedia: Explainable artificial intelligence <https://en.wikipedia.org/wiki/Explainable_artificial_intelligence>

[37] A version of the article 22 requirements were in fact present in the GDPR's predecessor, the data protection Directive of 1995, where it was to some extent ignored. It is the growth of AI in recent years that has made the GDPR version more prominent.

AI to be explainable where personal data is concerned, but it makes it clear that explainability will be a significant legal requirement for many RaC applications. The EU's proposed AI Act is broader than the GDPR in some respects, including in that does not require personal data to be involved before it applies. There is a substantial literature on these issues in the EU.[38]

Legal requirements of explainability are not restricted to the EU. Many other countries now have such requirements (or related controls on automated decision systems). At least 25 jurisdictions outside the EU[39] have such controls, influenced by the EU's DPD or GDPR, but often with different terms. So more than 50 countries now have some controls on automated decision systems.

Many other types of legal provisions, not just those in data privacy laws, may have the effect of requiring explainability where RaC applications are used. Guihot and Bennett Moses survey the provisions under Australian administrative law that could have such a result.[40] There are dozens of provisions in Australian federal law which authorise the making of decisions by automated means, but they then deem those decisions to have been made by a human decision-maker (for example, the Secretary of a government Department), thus bringing into play all the requirements of federal law relating to the making of decisions.

> 'What administrative law requires is that the reasons for a decision explain the outcome of a particular decision in accordance with the law under which the decision was made. For example, explanations may need to note if a statutory requirement for a particular benefit was not met or if a request falls outside an agency's statutory authority. A decision-maker may also need to describe how they have exercised their discretion when weighing several relevant factors.'[41]

They suggest that the law may mean such explanation should be consistent whether created by a human or an automated system, in which case automated systems 'must pass a kind of Turing test'.[42] They also suggest that Australia's anti-discrimination laws may require protections distinct from the giving of explanations, requiring testing of 'data driven inferencing' (systems based on machine learning) sufficient to ensure that it does not result in indirect discrimination.[43]

Such legal requirements will differ a great deal between countries, as will their implications for what counts as explainability in each country. Many countries outside Europe are considering regulating AI. An early proposal for national legislation is a 2021 Report by Australia's Human Rights Commission,[44] which makes 23 recommendations for the regulation of AI, almost all involving legislation. The Australian proposals include the following:

- An AI Safety Commissioner should be established by federal legislation, independent of government, with powers that are primarily recommendatory,
- For (federal) public sector use of any AI-informed decision-making system, 'human rights impact assessments' (HRIAs) should be required, focusing on compliance with international

---

[38] For examples, see Ebers op cit; P. Hacker and J-H. Passoth, Jan-Hendrik 'Varieties of AI Explanations under the Law. From the GDPR to the AIA, and Beyond' in: Holzinger, Goebel, Fong, Moon, Müller and Samek (eds.), *Lecture Notes on Artificial Intelligence 13200: xxAI - beyond explainable AI*, Springer, 2022, < https://ssrn.com/abstract=3911324>; G. Bar 'Explainability as a legal requirement for Artificial Intelligence' W*omeninAI - Medium* 27 November 2020 <https://medium.com/womeninai/explainability-as-a-legal-requirement-for-artificial-intelligence-systems-66da5a0aa693>

[39] China, Macau, Philippines (Asia - 3); Ghana, South Africa, Morocco, Kenya, Uganda, Algeria (Africa - 6); Brazil, Argentina, Uruguay, Peru (Latin America – 4); California (1); Albania, Turkey, Ukraine, Azerbaijan, Bosnia & Herzegovina, Serbia, Russia (Europe – 7). Only the 'Top 50% by GDP' of jurisdictions with data privacy laws have been considered, so that the actual number will be higher. The UK, and the three Channel Island jurisdictions should also be added.

[40] M. Guihot and L. Bennett Moses *Artificial Intelligence, Robots and the Law* (Lexis Nexis Australia , 2020), pp 166-175..

[41] Guihot and Bennett Moses, p. 169.

[42] Guihot and Bennett Moses, p. 169, citing Palmer Olsen on the 'Turing test' analogy.

[43] Guihot and Bennett Moses, pp 170-174.

[44] Australian Human Rights Commission (AHRC) *Human Rights and Technology Final Report* (2021), plus summaries <https://tech.humanrights.gov.au/downloads?mc_cid=f780633f2f&mc_eid=5a2fe75aaf>

human rights obligations, automation of discretionary elements, review by human decision-makers, and legislative authorisation. Legislation should require affected individuals to be notified. It should prohibit administrative decision-making which 'cannot generate reasons or a technical explanation for an affected person'. It should guarantee merit review appeal rights for any AI-informed decision.

- For private sector use of any AI-informed decision-making system, HRIAs are only recommended, with guidance from the AI Safety Commissioner. However, legislation should require that individuals be informed of the use of AI systems in situations affecting legal or other significant rights. Law should confirm that legal liability is regardless of how decisions are made, and that court or regulatory orders for production of information will apply no matter whether use of AI technologies make compliance difficult.

These are strong recommendations,[45] but there is no indication yet that legislation will follow. The Commission also recommended further research on explainable AI:[46]

> First, it is good practice to provide reasons for decisions that affect a person's legal or similarly significant rights, regardless of the status of the decision maker and even where there is no legal requirement to provide reasons. The Commission recommends further research on explainable AI, and expert guidance from government on how to provide reasons for AI-informed decisions.

These Australian proposals for legislation are only a small sub-set of those proposed by the EU, but it would not be surprising if there was some greater degree of convergence by the time both jurisdictions (and others) have completed their legislative processes.

We can conclude that requirements of explainability exist in some jurisdictions independently of data protection laws, that requirements influenced by GDPR Article 22 are starting to be found in many countries (at least 50), and that broader legislative requirements, such as in the EU's proposed AI Act, many also start to be enacted. Any RaC applications related to law must factor in the permanence of explainability requirements.

3.3. What should we look for, as XAI in RaC?

What will be acceptable as explainability differs in RaC from ML and other forms of AI. As the subjects of decisions with significant legal and social effects, we all require explanations that are legally sound, and based on permissible legal sources. It is not good enough for an explanation to be subjectively convincing, or to count as explainable in other contexts.[47] While it is easy to have this intuitive notion of what 'explainability' requires, it is still a developing concept in relation to RaC, and related, overlapping, terminology such as 'trustworthy systems' is also often used. We will now review some of explanations of 'explainability' put forward in the field of RaC.

Andrews[48] argues that 'trustworthy systems', in relation to RaC, must be considered in light of five key questions which are directed mainly at RaC system developers, or at least at organisational users:

1. How would you audit process/decisions in real time?
2. How would a citizen appeal a decision?
3. What are your checks, balances and oversight?
4. How would you know whether something is having a negative effect on people?
5. What does the public need from you to be considered trustworthy?

---

[45] Another recommendation, not relevant here, is that legislation should regulate uses of facial recognition and other biometric technologies. Until it is enacted, Australian governments should impose a moratorium on their use.

[46] AHRC *Human Rights and Technology Final Report,* p59.

[47] Guihot and Bennett Moses, pp 151-159.

[48] Pia Andrews, PPT headed 'Trustworthy systems', *Rules as Code 2.0. Global Plenary*, March 2022 (Australasian Society for Computers & the Law)

These are valuable questions that any organisation implementing RaC should ask, and the answers to them will often require explainability, but they do not tell us what explainability is.

de Sousa argues[49] that building trustworthy systems using coded rules requires that a system should meet four criteria to be considered trustworthy, each of which is then explained:

> "***Transparent*** - your rules should be visible.; ***Traceable*** - the steps in the decision-making process should be explainable/auditable.; ***Accountable*** - you should stand by the decision made by the system as a valid decision of your organisation.; ***Appealable*** - the subject of the decision should be able to seek a review of the decision (for example, regarding an error of fact, or if they believe a rule has been incorrectly applied)."

In similar vein, the OECD's Observatory of Public Sector Innovation (OPSI) published 'Cracking the Code' (2020), [50] intended as a guide to government implementation of RaC. They advocated six principles that they say should be observed in 'operationalising RaC': [51]

> ***Transparency*** – the rules generated by RaC projects, and the processes and thinking involved in their creation, must be transparent for end-users and citizens. This could be achieved by directly exposing coded versions of rules (which would make rules more transparent for technical experts), but also indirectly through front-end applications (which allow citizens to assess their own circumstances in relation to the rules). …
>
> ***Traceability*** – strongly related to the goal of transparency, the thinking and decisions underpinning the generation of machine-consumable rules should be clearly documented and understandable. … Traceability requires that the coded rules isomorphically reflect the original rules,
>
> ***Accountability*** – … For companies to base their regulatory compliance on an official set of machine-consumable rules, for example, they must have certainty that the rules are correct and consistent with the relevant human-readable counterparts. The organisation or entity publishing the rules should therefore be willing to assure the correctness of the rules. This would require that the government is accountable for the coded rules if errors are made.
>
> ***Appropriateness and Appealability –*** Appropriateness requires that consideration be given to the question of if a RaC approach is suitable for a given area or problem. This will include determining if generating machine-consumable rules will create value, as well as if available technology solutions possess the required capability. Of course, errors will inevitably arise in the coding of rules. Accordingly, there also must be mechanisms that allow the coded version to be corrected or appealed. Further, to the extent that a jurisdiction chooses to treat the coded version as having the force of law, the importance of mechanisms that allow the subject of the decision to seek a review (undertaken by a human actor) will rise.

They also advocate ***Availability and Interoperability*** ('Rules should be published openly and with mechanisms that enable their consumption by third parties.') and ***Security*** (particularly where coded rules are authoritative).

We support these principles, parts of which (isomorphism, appropriateness) reflect criticisms we made of the OECD draft report.[52] However, these 'OECD principles' still need to be strengthened in various ways:[53]

- It is a mistake to think that ***transparency*** should be limited to technical experts. It is much preferable if the coding language is so close to natural language that ordinary people (or at least ordinary lawyers) can read and understand what it is doing. But the application when running should also explain to the user, in natural language, what it is doing,

[49] Tim de Sousa *Rules as Code Handbook,* section 3, 23 February 2021 <https://github.com/Rules-as-Code-League/RaC-Handbook/wiki/3-Building-trustworthy-systems-using-coded-rules>

[50] OECD OSPI (J. Mohun and A. Roberts) *Cracking the Code: Rulemaking for humans and machines*, OECD Working Papers on Public Governance No. 42, 12 October 2020 <https://oecd-opsi.org/publications/cracking-the-code/>

[51] Ibid, Chapter 8.

[52] For a critique of their draft report, see G. Greenleaf, A. Mowbray and P. Chung, 'Strengthening Development of Rules As Code: Submission to the OECD's OPSI on Cracking the Code' (June 23, 2020). <https://ssrn.com/abstract=3638771>

[53] Ibid

- The main purpose of ***traceability*** is that the coded rules can be audited by third party experts (not the authors of the code), so that they are able to be traced back to the 'original rules' (in legislation or elsewhere).
- Trust in coded rules can also arise from the assurances of third party auditors, as discussed above under 'traceability', not only from government assurances, and it is important that this type of '***accountabilty***' be included in a full description.

We consider that two other aspects of explainability are necessary for RaC to work at its best:

- ***Sustainability*** is more likely when development which maximises the ability of organisations to write, understand and update codebases from their own legal and administrative staff members, with minimal reliance on computing staff for development and maintenance.[54]
- ***Access to legal sources*** – In most legal applications, encoded rules cannot by themselves deal with every situation that may arise in practice, so interpretation by users of the terminology used in rules (or in questions they generate) is necessary. Where possible, rules should be linked to the legal sources on which the rules are based, to facilitate user interpretation.[55]

3.4. Conclusions: Desirable XAI in RaC

We can now derive a list of desirable features of explainability in relation to RaC:

(i) The software running RaC apps should be freely accessible, preferably open source, so that technical developers can check that it operates as claimed (*Transparency*).

(ii) The language in which RaC apps are written should be humanly-understandable ('quasi-natural language' or 'English like'), and therefore able to be written and maintained by those providing legal advice (*Transparency*).

(iii) Coded rules have as close to an isomorphic relationship as is achievable to the sources on which the coded rules are based (*Traceability*).

(iv) Codebases for apps should be publicly available, and re-usable by others (*Availability*).

(v) When apps run, they should be able to explain their questions, reasoning and conclusions while the app runs (*Transparency*). Ideally, these explanations should be generated from the logic and text of the app (not from pre-formulated separate answers) (*Sustainability*).

(vi) All of these run-time explanatory features should be *linked to the legal sources* on which they are based.

(vii) When legal authorities make codebases available, they should be *accountable* for the codebase being an accurate counterpart to its legal text equivalent.

4. Representing RaC to Facilitate Explanation

In order to facilitate explanation in systems built around RaC that are based on legislation or other rules, it is important to make the relationship between the elements comprising rules explicit. Explainability for RaC involves being able to describe how these elements relate and how they affect each other.

Legislation is generally comprised of parts, divisions, sections and various types of subsections which are used to organise a set of considerations and outcomes. Each of these considerations and outcomes is an element.

[54] Graham Greenleaf, Andrew Mowbray and Philip Chung 'Building sustainable free legal advisory systems: Experiences from the history of AI & law' (2018) 34(1) *Computer Law & Security Review 314,* particularly at [3.6].

[55] See 'Building sustainable free legal advisory systems', above cited, particularly [3.11] – [3.14].

For example, where a section contains three sub-sections forming conditions, the explanation for the section applying or not applying can be found in whether or not all or some of the sections are applicable. Similarly, during execution, the application of a sub-section is being explored then the explanation for why this is happening is that this is relevant to a parent section.

Whilst each element may require interpretation or explanation, this is outside the scope of what is represented in an RaC codebase.

When representing Rules as Code, elements may be substantive or structural.[56] The structure and relationships between elements are made explicit to support the ability to provide explanation for how non-structural, that is, substantive elements affect outcomes and are affected by other elements.

Consider for example, a RaC implementation of a traditional legislative section such as s114, Patents Act 1990 (Cth). The text of the section is:

> **PATENTS ACT 1990 - SECT 114**
>
> **Priority date of claims of certain amended specifications**
>
> (1) This section applies if:
>
> (a) a complete specification has been amended; and
>
> (b) the amendment was not allowable under subsection 102(1); and
>
> (c) as a result of the amendment, a claim of the amended specification claims an invention that:
>
> (i) was not disclosed by the complete specification as filed in a manner that was clear enough and complete enough for the invention to be performed by a person skilled in the relevant art; but
>
> (ii) is disclosed in that manner by the amended specification.
>
> (2) If this section applies, the priority date of the claim must be determined under the regulations.

In order to represent this for RaC purposes that would facilitate explanation, the structural and substantive elements must be recorded. Conceptually, this section can be regarded as the following directed graph.

[56] See Mowbray, Chung and Greenleaf "Representing legislative Rules as Code: Reducing the problems of 'scaling up'"

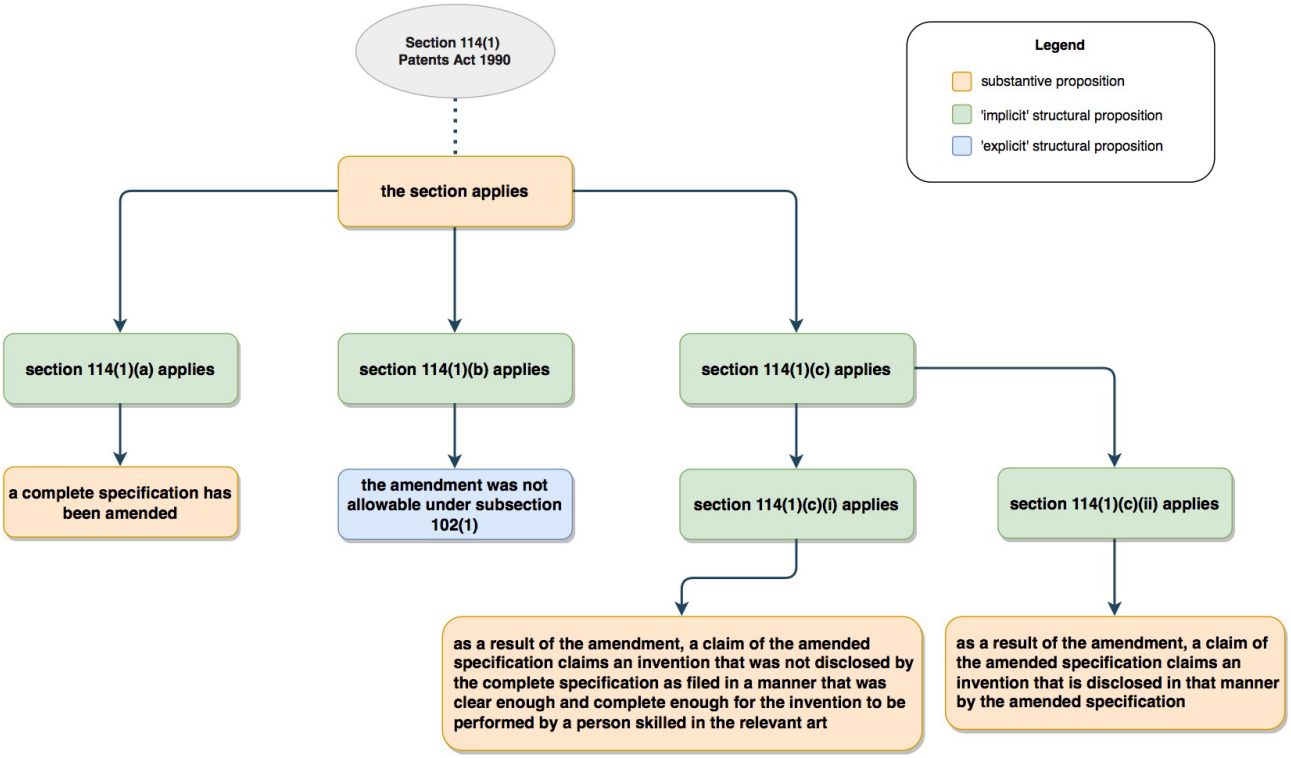

Figure 1 Section 114(1) Patents Act 1990 (Cth) represented as a directed graph of propositions

The graph organises substantive and structural elements in a way which explains relationships between elements and how elements can be decided and affect each other. For example, the section will apply if the structural elements s114(1)(a), (b) and (c) all apply. Similarly, s114(1)(a) depends upon the substantive element that "a complete specification has been amended" is true. Explanation for why this substantive issue is being considered will be that its parent element "s114(1)(a) applies" is being considered, and the reason that this is being considered is because s114 itself is being considered.

Whilst the above example is detailed and may appear complex, the fundamental point is a simple one, that is for the operation of a set of rules to be explainable, it is necessary to capture the way that they work. A codebase needs to reflect how elements contained in rules affect each other.

5. DataLex: RaC meets XAI, in the public domain

DataLex is an applications development environment created by the university-based Australasian Legal Information Institute (AustLII) which is suitable for developing RaC codebases and applications.[57] The system uses a language called "yscript" which facilitates the declarative representation of rules using a quasi-natural-language codebase syntax (ie one resembling English).

For RaC codebases, human rules (such as sections contained in legislation or regulations that are being encoded) are generally represented isomorphically (that is, human rules are mapped on a one-to-one basis to yscript rules). Each rule element (be it a premise or conclusion) is represented as a "fact" which is often in the form of a proposition. Rules set out the relationships between facts (for example, "if some fact applies as well as another fact then some conclusion can be drawn").

[57] For an overview of the DataLex components, see A. Mowbray, G, Greenleaf, and P. Chung, *Law as Code: Introducing AustLII's DataLex AI* (November 16, 2021). UNSW Law Research Paper No. 21-81, <https://ssrn.com/abstract=3971919>.

There is no separate coding of what a codebase should "do", nor for specific explanations or of other system dialog. All interactions are generated automatically from the facts contained in the rules, in dialogues generated 'on the fly' when the system is in operation.

The DataLex approach does not require the involvement of software experts and codebases can be created directly by lawyers or legal drafters.

Applications and codebases can be collaboratively developed and maintained within the AustLII Communities environment and integrate with AustLII using automated hypertext links to Australian legislation and cases.[58] Access to the DataLex development environment and documentation are available from the DataLex website.[59]

RaC based systems built using the DataLex system meet the seven desirable features of XAI discussed in the previous section as follows:

5.1. the yscript language is open source

The yscript interpreter (the software that runs apps written using the yscript language) and yscript library are available as open source[60] under an Affero GPL licence.[61]

5.2. yscript uses a quasi-natural language representation for applications

yscript[62] uses a quasi-natural-language 'English-like' syntax, which is easy to learn and use, supports declarative and imperative coding, and produces natural English dialogs (consultations). While yscript is a flexible general purpose language, it is particularly useful for representing legislation and other rules which are comprised of a structured set of propositions.

One of the central aims in the development of *yscript* was to develop a form of representation that looked as much like natural language as possible. The language syntax manages to almost entirely avoid the use of symbols which are the principal structural elements of most programming languages. This was done partly to make it easier to write code for non-programmers, but also to make the code more transparent. Even if someone cannot write code in yscript, they can probably understand what it is doing and possibly even comment upon whether it accurately encapsulates anything from the real world (such as the text of legislation) that it is meant to reflect.

5.3. yscript encourages isomorphic representation

The rule-based structure of yscript encourages and supports isomorphism of real-word rules (particularly legislation) into yscript code, important for transparency, explanation and maintainability). It also makes it a lot easier to build applications, and of equal importance, it allows for simpler maintenance of the code when source legislation or other rules change, and for legal experts to audit the accuracy of the code without being computing experts.

An extract from the Australia's Foreign Relations (State and Territory Arrangements) Act 2020 (Cth) codebase is shown in Figure 2.

---

[58] DataLex Community web pages <http://austlii.community/wiki/DataLex/>.

[59] See http://datalex.org

[60] The yscript source is available via the main DataLex page < main http://datalex.org/> under "Source". The link is: https://datalex.org/src/ys/ys-latest.tar.gz

[61] An explanation of the Affero AGPL licence is at: <https://www.gnu.org/licenses/why-affero-gpl.html>.

[62] An earlier version of the *yscript* language was originally developed for the expert systems shell *ysh*. Prior to being integrated into AustLII's DataLex platform, yscript was also used as the language and code interpreter for a system called wysh (short for "web-ysh").

```
RULE Section 10 - Core foreign arrangements PROVIDES
the arrangement is a "core foreign arrangement" under section 10(2) ONLY IF
    the arrangement is a "foreign arrangement" under section 6 AND
    the Australian entity is a "core State/Territory entity" under
    section 10(3) AND
    the non-Australian entity is a "core foreign entity" under section 10(4)

RULE Section 10(3) PROVIDES
the Australian entity is a "core State/Territory entity" under section 10(3)
ONLY IF
    section 7(a) applies OR
    section 7(b) applies OR
    section 7(c) applies
```

Figure 2 **Extract of a DataLex codebase written in yscript**

yscript does not require questions and explanations to be written separately from the rules that make up an application. Questions and explanations (responses to 'Why?', 'How' and 'Conclusions') are generated on the fly by the yscript interpreter from the rules when the app is running. There is therefore no textual 'baggage' which must be written (and maintained) in addition to its rules.

It is intended that *yscript* applications can be directly created and maintained by lawyers.

In formal terms, yscript code consists of *rules* that deal with *facts*. Facts are expressed in their plain English-language form. Individual rules can be imperative but more often are declarative and describe the relationships between facts. Once a rule is being evaluated, other rules that can help determine a value for required facts are automatically executed. Rules are used in a goal-oriented fashion to determine values. Each time that a new fact becomes known, rules are used to check if other fact values can be derived. When required, rules can be specifically called like procedures or functions in other languages.

5.4. Codebases in yscript are generally publicly available

It is desirable that apps written in yscript, so that they can be run with the interpreter, should be licensed under some form of open content licence, and made available for any user to read,[63]

5.5. Explanatory features of yscript apps, when running

When yscript code is interpreted and run, it results in a dialog or *consultation*. A series of questions are asked, and conclusions are made. During the consultation, the user can interrogate the system as to why questions are being asked (Why?), to explain how conclusions have been reached (How?), and to delete facts so that all conclusions depending on the fact are re-evaluated.

The DataLex user interface uses the DataLex software and codebases, the linkages provided by the Communities environment, and user input, to provide legal advisory systems in operation.

[63] The yscript code for many apps is available at <http://datalex.org>.

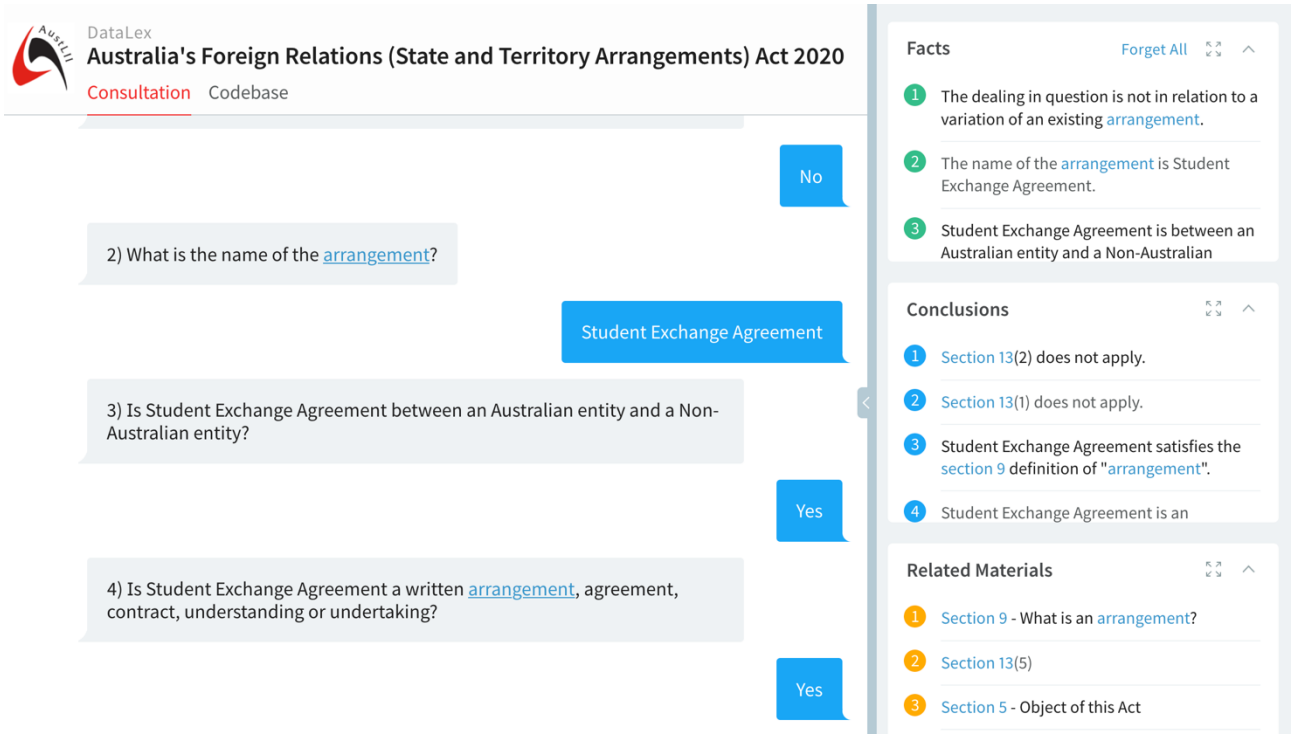

Figure 3 DataLex user interface features: Consultations, Facts, Conclusions, Related Materials

From the above user interface extract, and further interface extracts below, eight elements of DataLex's 'explainability' are shown:

1. Questions, Facts, Conclusions, and Reports are all generated from the codebase plus user-provided facts, are in an understandable form ***('English-like')*** and are available on screen at all times.
2. Facts previously provided by the user (shown under Facts on the right side) can be deleted (***'Forget?'***), by selecting the number of user-provided fact, and questions are then re-asked to re-establish a value for that fact.

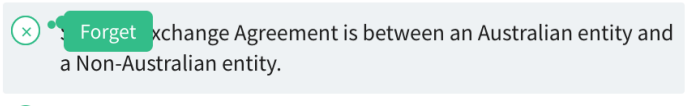

Figure 4 DataLex 'Forget' a user-provided fact during consultation

3. Reasons for why a Question is being asked can be requested ***('Why?')***, with the reasons being generated on the fly from the relevant rules.
4. Conclusions are shown on the right-hand side. Selection of a numbered conclusion results in a ***'How'*** explanation of that conclusion being presented, as shown in Figure 5.

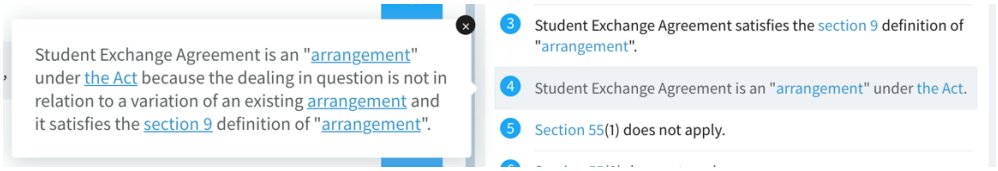

Figure 5 DataLex 'How' explanation during consultation

5. The system also uses all information available to it, from the codebase and user-supplied facts, to suggest other relevant ***Related Materials*** which it extracts from the whole AustLII system and displays the most relevant results.

6. Users can test a hypothetical answer to a question through selection of '***What if?'*** (that is, 'what happens if I answer this way?'). When *What If?* is turned off, the original question will be asked again.

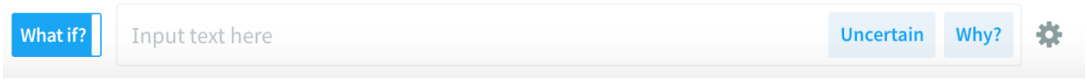

Figure 6 DataLex 'What if?' option selected during consultation

7. The consultation can be set in ***Verbose*** mode, so that each of the rules that are fired is displayed as they are fired, enabling the user to see the relationship between the rules and the interface.

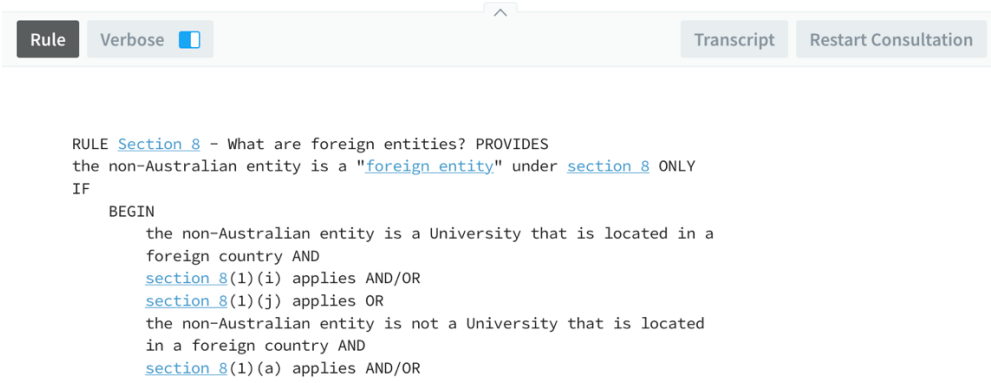

Figure 7 DataLex 'Verbose' mode enabled showing current rule being considered

8. When a session completes, a ***Report*** is generated to give an answer to the original goals set for the consultation, and to explain why this answer follows from the user-provided information given during the consultation. These reports may be quite lengthy.

These features are important to achieving the objective of 'explainable AI'. They demonstrate the value of an 'English-like' interface by showing how various and complex the interface to an AI system can be. *Why?* and *How?* explanations – and *What If?* – show how the interactions in a consultation can be constantly explaining aspects of the system's reasoning to a user. The explanations available for each Conclusion, and their being amalgamated into a lengthy Report at the end of the consultation, both generated from the rules that have been fired, and the user-provided answers to questions.

5.6. Explanations to link to source

The DataLex interface automatically links all references to primary materials (legislation and caselaw) to AustLII where available.

5.7. Using ylegis to make law and code identical

The Hairdressers Act 2003 (NSW) example also illustrates two ways of representing Rules as Code in DataLex. One way is to code existing legislation in yscript as part of a conversion process (the majority of current example codebases are written in yscript). Another approach is to write the legislation in a natural language format that can also be executed as code (the ylegis format). A pre-processor program (ylegis) takes a section of legislation (or multiple sections) and converts it automatically into a 'first draft' of yscript code for those legislative provisions, which can immediately be run by the yscript interpreter.[64]

[64] For details and examples, see A. Mowbray, P. Chung, and G. Greenleaf, 'Representing legislative Rules as Code: Reducing the problems of 'scaling up'' (December 9, 2021). <https://ssrn.com/abstract=3981161>

The ylegis format can be used to write legislation in a form that is close to existing conventional legislative drafting but requires relationship between separate propositions be defined by a formal set of connectors (such as 'and:' and 'or:') and these operators are used to connect subsections and subclauses each of which contains only a single proposition.[65]

Using s4(1) of the Hairdressers Act 2003 (NSW) as an example, the following three figures show the original drafting (Figure 8), the section being coded in yscript (Figure 9) and the section rewritten in the ylegis format (Figure 10).

4 When is an individual "qualified to act as a hairdresser"?

(1) For the purposes of this Act, an individual is
"qualified to act as a hairdresser" if any one or more of the following applies to the individual--

(a) the individual has been awarded an authorised qualification by a registered training organisation,

(c) a determination has been made under section 37 of the *Apprenticeship and Traineeship Act 2001* that the individual is adequately trained to pursue the recognised trade vocation of hairdressing (because the individual has acquired the competencies of the recognised trade vocation),

(d) the individual has at any time held, or been taken to have held, a licence under Part 6 (Regulation of the hairdressing trade) of the *Shops and Industries Act 1962* , other than a licence limited to carrying out beauty treatment only.

Figure 8 Section 4(1) of the Hairdressers Act 2003 (NSW)

```
RULE Section 4 - When is an individual "qualified to act as a hairdresser"
PROVIDES
SUBRULE Section 4(1)
SUBRULE Section 4(2)

RULE Section 4(1) PROVIDES
the hairdresser is qualified to act as a hairdresser ONLY IF
    section 4(1)(a) applies OR
    section 4(1)(c) applies OR
    section 4(1)(d) applies

RULE Section 4(1)(a) PROVIDES
section 4(1)(a) applies ONLY IF
    the hairdresser has been awarded a hairdressing qualification AND
    the qualification was awarded by a registered training organisation AND
    the qualification is an "authorised qualification" under section 4(2)

RULE Section 4(1)(c) PROVIDES
section 4(1)(c) applies ONLY IF
    a determination has been made under section 37 of the Apprenticeship and
    Traineeship Act 2001 that the hairdresser is adequately trained to pursue
    the recognised trade vocation of hairdressing (because the hairdresser has
    acquired the competencies of the recognised trade vocation)

RULE Section 4(1)(d) PROVIDES
section 4(1)(d) applies ONLY IF
    the hairdresser has held, or been taken to have held, a licence under
    Part 6 (Regulation of the hairdressing trade) of the Shops and Industries
    Act 1962 AND the licence was limited to carrying out beauty treatment only
```

Figure 9 Section 4(1) of the Hairdressers Act 2003 (NSW) in yscript format

[65] See Mowbray, Chung, Greenleaf "Representing legislative Rules as Code: Reducing the problems of 'scaling up'"

```
4. When is an individual "qualified to act as a hairdresser"?
(1) An individual is qualified to act as a hairdresser if—
    (a) the individual has been awarded an authorised qualification by a
        registered training organisation; or
    (c) a determination has been made under section 37 of the
        Apprenticeship and Traineeship Act 2001 that the individual is adequately trained
        to pursue the recognised trade vocation of hairdressing (because the
        hairdresser has acquired the competencies of the recognised trade
        vocation); or
    (d) the individual has held, or been taken to have held, a licence under
        Part 6 (Regulation of the hairdressing trade) of the
        Shops and Industries Act 1962 and the licence was limited to carrying out beauty
        treatment only.
```

Figure 10 Section 4(1) of the Hairdressers Act 2003 (NSW) in ylegis format

When drafting legislation using the ylegis approach, the resulting text is both legislation and code.[66] This means that if a piece of legislation is written using the ylegis format, an intermediate yscript version will be generated automatically during execution on which explanations will be based without having to code a separate version of the legislation in yscript.

This is a different approach to explainability: the legislation *is* the code. Legislation drafted in ylegis format has the same desirable features of explainability as legislation represented in yscript (as set out in the following section).

5.8. DataLex apps as explainable AI

Explainability is demonstrated in the numerous applications written in *yscript* that are on the DataLex web pages,[67] including examples on legal subjects such as the Mandatory Bargaining Code (Cth), NSW Community Gaming Regulation, the Modern Slavery Act 2018 (Cth), and the NSW Hairdressers Act. The yscript code for each application can be found on the DataLex site, and the applications can be run ('Consultations'). Users can develop and run their own test applications using the *DataLex Application Developer Tools*.[68]

In summary, we can now check which of the seven desirable features of explainability (see section 3.4) are provided in DataLex's implementation of RaC:

> *Transparency* – yscript interpreter software is open source; yscript language for app development is humanly-understandable ('quasi-natural language' or 'English like'); When yscript apps run, they are able to explain their questions, reasoning and conclusions while running.
>
> *Traceability* – yscript rules can have as close to an isomorphic relationship as is achievable to the sources on which the coded rules are based.
>
> *Availability* – yscript code bases for apps should be publicly available, and re-usable by others (and are for many examples).
>
> *Sustainability* – Run-time explanations in yscript apps are generated from the logic and text of the app (not from pre-formulated separate answers or questions).
>
> *Links to the legal sources* justify all run-time explanatory features.

The same qualities apply to legislation written in ylegis format.

---

[66] See <http://austlii.community/foswiki/DataLex/HairdressersActKB>.

[67] DataLex web pages <http://datalex.org/>

[68] *DataLex Application Developer Tools* <http://datalex.org/dev/tools/>

One desirable feature ('accountability') requires an answer to the question of whether legal authorities that make codebases available also make themselves *accountable* for the codebase being an accurate counterpart to its legal text equivalent. It is a question for governments, not RaC developers.

6. Conclusions

We have argued that the long-term interests of justice require that a significant portion of RaC tools and applications are available as part of the public domain, a shared resource for the 'have-nots' of the legal system. This is necessary for the future development of 'free access to law'.

The transparency associated with Explainable AI is equally necessary for systems based around RaC. This is the case regardless of whether applications using RaC embody AI components or approaches.

'Explainability' is going to be a permanent legal requirement globally for both RaC and AI systems, or at least for those involving personal data.

We have shown that DataLex's implementation of RaC, and the tools it provides, can provide a high level of explainability for RaC applications. Explainability is a core aim of the approach that AustLII has adopted in building RaC codebases and applications using DataLex. Representing human rules following the DataLex methodology as outlined incorporates the necessary information to explain how these rules work and why they produce the outcomes that they do.

DataLex and the yscript language have been used to build a relatively large number of RaC based applications that demonstrate the effectiveness of the underlying approach.

Current work is focusing on using the system to deliver practical outcomes for a number of use cases. In part, this has to do with making the building of codebases from existing rules more efficient as well as building better interfaces for applications.

Looking into the future however, the best way to build RaC systems (using DataLex or otherwise) is to create rules in a form that are directly machine consumable. The ylegis "language" provides the opportunity to explore the viability of creating legislation, regulations and other rules that are directly usable both by a human and machine audience. Experiments need to be done to see how far this can be taken in practice and whether it is possible to alter current drafting practices to better suit the requirements of modern rule-making.

Rules which can also function directly as code inherently incorporate their own explanation and justification. Writing laws in this way has the potential not only to produce rules that can be used more flexibly, but also to deliver more certainty and transparency which will enhance the operation of the rule of law.