

## Rules as Code – 7 Levels of Digitization

Regan Meloche - Dec 16, 2020

This is a brief summary of Rules as Code – Seven Levels of Digitisation by Huang Mingrong, which outlines a conceptual model for concretely defining what is meant by “Rules as Code”.

The term “Rules-as-Code” is used in many different contexts and it can have different meanings depending on how it is used. This is an attempt to establish a common frame of reference around the meaning of the term. In the loosest sense, the term may refer to *any* encoding of any type of rule, for example a simple *if* statement that checks if an age is under a certain threshold.

Technically this is an encoded rule, so this can lead to confusion around the term. This document establishes more rigid guidelines around what is meant by “Rules-as-Code”, so that it can be used more precisely and communicated more clearly.

### **Level 1: Digital First Steps**

This level refers to rules that have been scanned and digitized by a third party. The text is made searchable and tagged for structure. It could also include rules published in HTML or other digital format, such as one specialized for legal documents. Ideally this could even be an open standard, such as Akoma Ntoso.

### **Level 2: Digital Applications and Products**

At this level, the rules are in their own software package as an authoritative source, which may be used by other public and private sector entities. The logic of the rules may be directly translated to code (e.g. as a series of *if* statements), but an important note here is that if the rules change, then the code needs to be changed. Since it is built into the code, it requires someone with programming experience to work within the system.

An example that embodies certain aspects of level 2 is OpenFisca.

### **Level 3: Declarative Rules**

At level 3, the rules are implemented in their own declarative application layer that is dedicated to operationalizing the rules. An important distinction between levels 2 and 3 is that at level 3, a “non-developer” is capable of understanding the rules specification. If the rules change, then it is a simple matter of deploying the logical rule specification, rather than needing to re-deploy the entire application. This is the stage where a cross-disciplinary approach between rule writers and technologists can be valuable.

The rules themselves may be specified by a third-party syntax, or more ideally, by an open standard such as YAML. The separate rules engine can be deployed independently and serve as an API for multiple user-facing applications, and also for general public consumption.

### **Level 4: About Ontologies**

Level 4 moves into a more sophisticated area of rules – those concerned with constitutive rules, which asks questions about defining terms, such as:

- What entity counts as a “person”
- What does a “business day” in a given country refer to specifically

This ontology may be implicit, where it is assumed that everyone involved has a shared understanding of common terminology. This approach doesn’t scale to more nuanced rules, so at some point, the ontology may need to be expressed in some form of documentation. Further steps include to specify the ontology in any machine-consumable syntax or an open standard.

## **Level 5: Natural Language Generation**

Level 5 is the first to adopt a fully “digital-first” approach. It moves beyond the specification about the digital system and into the culture of policy creation itself. At this level, machine-consumable code is created in parallel with natural language rules. This creates the concept of a “digital twin” of the written rules, and both may be seen as the authoritative source of truth.

## **Level 6: Tooling Automation**

At this point, we have largely moved into the realm of science fiction (for now), where we have automated systems that can fully rely on the formalized rules. An example of this may be an expert system that can be integrated into a word processor that can identify and correct any rule-specific errors.

## **Level 7: Universal Adoption**

Universal adoption and normalization of digitally encoded rules.

## **Application to a Policy Difference Engine**

This document offers a more concrete frame of reference for what it means to talk about a ‘Rules-as-Code’ system. As we explore the RaC space in our own context, it is important to come to a shared understanding of this terminology with various stakeholders and allies so that we can deliver a product aligned with their expectations. The definitions presented here can be used as a starting point, and as we progress through the project, we can validate these or refine them further.