

# Recursion: the key to creating unlimited complexity from simple rules

---

Roger L. Costello

July 12, 2014

I confess, I love recursion. Always have. Always will. It was love at first sight.

Lately I have been immersed in learning a data format called JSON and its auxiliary schema language called (appropriately enough) JSON Schema. JSON and JSON Schema are both specified in a recursive manner. For example, the JSON specification says:

1. A JSON document contains a `value`.
2. A `value` may be an array, an object, or a few other things.
3. An array is written like this: `[value, value, ... ]`

See the recursion? A `value` may be an array. An array contains a `value` (which may be an array, which contains a `value`, which may be an array, ...)

Wow!

Simple rules generating unlimited complexity.

By contrast, there is another data format called XML and its auxiliary schema language called XML Schema. XML and XML Schema are *not* defined recursively. Consequently their specifications (XML Schema in particular) are extraordinarily complicated. The XML Schema specification attempts to document every conceivable scenario. It does so, but at enormous mental expense: the XML Schema specification is over 300 pages of very dense descriptions written by a Ph.D. linguist in collaboration with 20 other people. Contrast with the JSON Schema specification which is 6 pages, written by 1 person. Furthermore, JSON Schema is more powerful than XML Schema!

Recursion: simple rules generating unlimited complexity.

Simply wonderful.

Some people say that God is a mathematician. I say that God is a master of recursion.