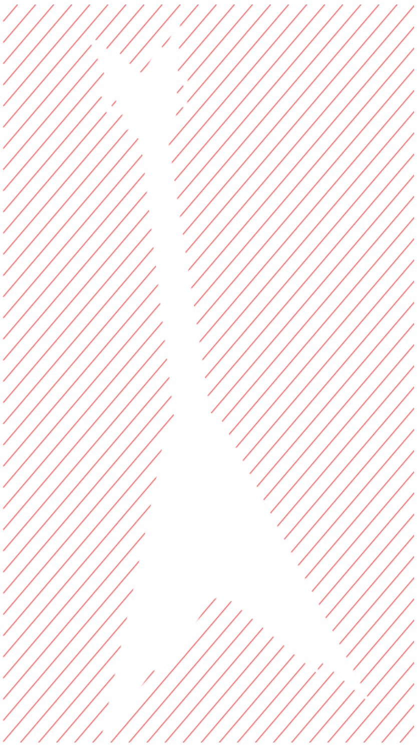
Finding what isn't there



During one of those US based procedural cop shows there is a scene where one of the principle stars comes into a room, looks at the blood splattered on the living room floor and notices a void. Of course the character is trying to image an object that is no longer present, but helpfully, for those amongst us viewers who can't picture it, a ghostly shape coagulates revealing that the void was caused by a Dime Razorback V electric guitar. This, of course, is the key to the whole case, the murderer is led away in handcuffs and our hero can put his sunglasses back on and stare moodily into the middle distance as the closing credits role.

In real life working out what isn't there is usually a bit more of a challenge. When we pick out that battered set of dominoes from the back of the cupboard we might know that there are two missing from how well they fit in their box, but we have to lay them out in sequence to identify exactly which two are absent. This is also what we do in data management, when reviewing the collection of workflow documents to see what needs working on next we list each one according to the task it describes and the data categories it covers. The gaps give us a list of candidates that still remain to be written.

The point here is an obvious one, like with the dominoes, we have to have an "overall structure" in mind if we are hoping to obtain a list of the things that aren't there. Without some kind of a model of what, ideally, we would like to have, it is hard to identify what is, so far, omitted. In E&P data handling creating that type of model is a bit more time consuming than sorting dominoes: we must categorise all the data we interact with; we must consider what steps an ideal "lifecycle" of technical data would include; we must worry about the various places that data comes from, goes to or sits in; we must agree what roles those repositories would ideally play; we must communicate what roles the various people involved should play (what are their responsibilities and interests); we must be aware how different groups within the organisation (different assets say) vary; and I'm sure there are various other concerns that, for the moment, I can't think of.

In a perfect world these issues would all be defined independently (or orthogonally to use a favourite word of mine). In reality they have a whole raft of intricate relationships, as a result defining any one of them constrains the way the others can possibly work. There is an example in the book where I consider the way that defining an ideal data lifecycle restricts the roles that repositories can play. Despite this "messiness" I am sure that systematically defining and documenting your key concepts individually is the most efficient way to start building a "model" of what is important for your data handling. In turn that has to be the best way to categorise the existing standards, policies, workflows, practice descriptions and other important documents you have. This makes what you have easier to find (because, if all the terms are well documented, everyone can use the same words when searching) but the major added bonus is that it should become easier to list the things you have not yet got.