means we are now in a position to simply describe the function of the device that we want and allow a compiler to determine how that should be instantiated in the instruction sets.

My desire to add a sun dial to my phone could be achieved through software or electronic or physical means or a combination of all — a compiler could work out that decision tree for me. This opens up a possibility for an entire new form of programming language that compiles down to physical, electronic and coding forms and where designers concentrate on describing the function of the thing and even object inheritance in the physical world. I called this theoretical programming language SpimeScript (*Point 6*) in honour of the marvellous book by Bruce Sterling on Shaping Things. This topic was my central theme of a talk I gave at Euro OSCON in 2006.

However, I had previously raised these discussions within the parent company and had become aware that whilst we might be able to make far future anticipations of change, they were increasingly built on layers of uncertainty and were increasingly unfamiliar and uncomfortable to others. The further we went, the crazier the ideas sounded and the more concerned people became. This itself creates a problem if you intend to motivate a team towards a goal. Hence, if I was going to choose a course of action, it needed to push the boundary but not too far so that it seemed like science fiction.

I was starting to feel uncomfortable with: -