Managing Complexity, October 3rd 2017:

Dinesh Verma on 'Complex to complicated'

During his speech at TNO-ESI's 2017 Symposium on 'Managing Complexity' professor Dinesh Verma seemed to alternate between the philosopher's, the linguist's and the engineer's approach to complexity and complication. It turned out that aspects from these fields together worked out fine to explain how the system architect got in his present situation and where he is at: he somehow needs to cope with complexity. "We need a lot more systems engineering institutes", Verma sighs.

"Why focus on complexity rather than simplicity?", Verma asks his audience rhetorically. "Because we tried simplicity ant it proves to be way harder. We therefore quickly went back to complexity again." Complexity, he says, increases in the entire universe – in nature and in society and not least in systems engineering. It is society's and the system architect's task to deal with complexity, as with the growing fragility that is linked to it. "The system engineer's toolbox for that, mainly dating from the 1940-1970 period, begins to fall short. There is a need for change, but a lot has to be overcome before a new toolbox will become mainstream."

Validation and verification

What has changed, is that electro-mechanic systems with a little bit of software have been replaced by systems in which it is all about software. This creates a fuzzy front end. To make software driven systems work almost without error, a change in information flow and decision making power is required. It is an illusion that there can be a 'right' answer and that 'unambiguousness' exists. The answer lies in validation and verification, early on and often repeated during projects.

The system architect has to aware of paradoxes. To name some: "The boundary's paradox: it is critical to have boundaries to projects, but these must remain permeable. The crowd paradox: systems of systems demand diversity, but it is essential that the sense of belonging to the whole does not get compromised. And the customer paradox: today's customer can get in the way of tomorrow's. How to deal with customers that don't evolve as quickly as systems allow?"

Complicated and complex

Verma thinks it is important not to confuse complication with complexity. "A jet engine built from tens of thousands of parts is complicated, but it isn't necessarily complex. Imaginary complexity can be overcome by gaining new knowledge, real complexity is complex by nature and mainly involves uncertainty."

In its core, complexity is about the question whether behaviour can be predicted. When this is possible, something can be complicated but not complex. For example: a hot or cold water system for a submarine consists of only five or six parts. But as the water behaviour cannot be predicted, it is complex. "When elements are involved that don't obey the laws of physics and when decomposition won't work, things get complex. So once you include humans in the equation, things become complex, for instance when you transfer authority from a system to humans in an autonomous car."

Abstraction and reduction

To deal with complexity, you need abstraction, reduction and homogenization. “More and better requirements used to lead to better design. That doesn't work anymore. Thousands of requirements could lead to worse design when flexibility gets lost in detail. To get a grip, proper scalability and reduction/filtering are needed. It is one of the priority challenges in systems engineering: to get to flexible designs that adapt and are resilient.”

In a gas, the behaviour of a single molecule cannot be predicted. Likewise, individual human behaviour is hard to predict. “But patterns in many gas molecules and groups of humans can be predicted. Looking at systematic behaviour allows for systems that respond in real time, which is essential to present cyber physical systems”, Verma concludes.

A new phenomenon is the system that learns and changes over time. “The challenge that comes with artificial intelligence for the system architect: how test such a system? And as systems become more complex, more and more disciplines become involved in their development. What should the system architect do to make the extreme diversity of present development teams work? Verma argues, with a wink but also a serious undertone, that some psychology input could come in handy, apart from engineering and philosophy.

Professor Dinesh Verma is professor of Systems Engineering at the Stevens Institute of Technology, Hoboken, New Jersey and Executive Director of the Systems Engineering Research Center (SERC) that unites the Systems engineering activities of 22 universities in the USA.