WHY IS EVERYONE TALKING ABOUT MBSE ANYWAY?

All my different sub-persons come across the term MBSE (model-based systems engineering): associate editor of Systems Engineering, fellow of INCOSE, professor in Norway, and researcher at ESI (TNO). These encounters have several different patterns:

- A large group of MBSE "advocates" look particularly concerned if a diagram does not use SysML.
- A large group of executives and developers hear the term and are afraid of missing out on something important.
- A group of older (?) professionals are disturbed by SysML and the high level of hype.

So, it is high time to bring some light into the darkness. Let's state some facts. The first simple fact is that for decades we have benefited from a wide range of digital tools for almost all our engineering work, such as requirements, configuration, and version management, software development tools, CAD-E, CAD-M, PLM, ERP, and the alphabet soup can go on for a long time. All those tools need a formalism and a representation to work. The result of engineering is that we create the technical production documentation, which is practically completely digital and largely formalised. Can anyone imagine developing a top-of-the line Intel CPU without extensive digital support?

Today's pain in developing complex systems is in various dimensions. A rather mundane problem is that although most information is digital somewhere, many steps in the entire development and product lifecycle require "manual" interventions; interoperability between many of the tools is still problematic. Part of the interoperability challenge is that many digital artifacts (models?) are discipline-oriented. How will all these artifacts form an integral, fluent solution? There is a clear need for integrating technology (with its formalism and representations) that streamlines the entire lifecycle.

A second fact is that complexity of the problem space causes another pain. Many systems that we make have a wide variety of stakeholders with an even wider range of needs and interests. These needs and interests go beyond technology, e.g. political, economic, social, environmental, and legal considerations are at play. Domain knowledge (defence, transportation, healthcare, etc.) plays a big role. The heterogeneity of stakeholders and considerations, in combination with ambiguity and fluidity of human interaction, requires communication, cooperation, influencing, and many other "soft" skills. This part of the pain is ill-served with formalism; it requires a broad spectrum of representations used in social interaction.

Our challenge is that we have to resolve both pains, in a way that both worlds function together. The digital interoperability will need something MBSE-like. Critical thinkers wonder whether the current SysML is a proper fit; that discussion goes beyond an editorial. The problem space complexity and the stakeholder interaction require a completely different paradigm of communication and visualisation, which I call conceptual modelling. Lastly, we need a way of working that is connecting both paradigms, continuously, from conception until decommissioning. This way of working is systems engineering.

Gerrit Muller
Professor of Systems Engineering, University of South-Eastern Norway,
and senior research fellow, Embedded Systems Innovations by TNO
gerrit.muller@gmail.com, www.gaudisite.nl