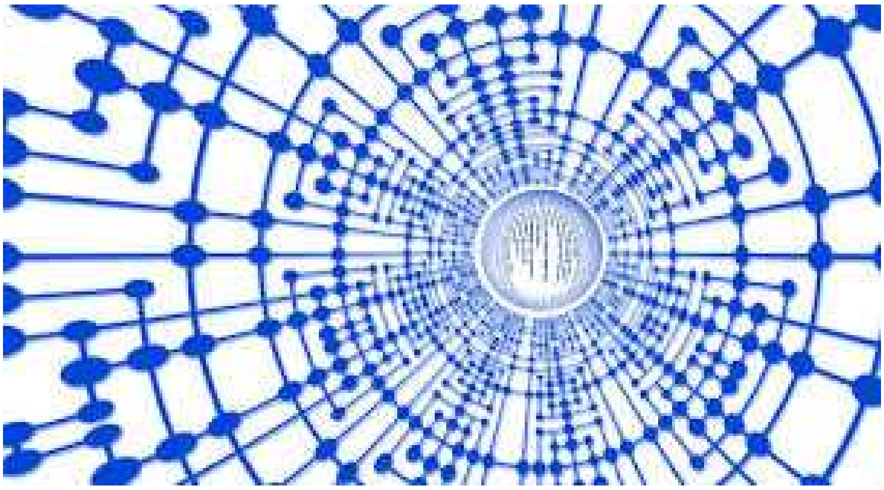
SYSTEM ENGINEERING: MATHEMATIZING IS NOT THE LAST WORD



Some System Engineering solution (method and tools) designers refer to mathematics as an authoritarian argument. In a domain where marketing prevails over any rational approach in the absence of any truly scientific foundation, decision makers usually do not question this point, if only not to sound stupid or ignorant.

What's the point? Actually, calculability, yet calculability is not synonym for refutability and physical predictability. In a System Engineering process, beyond the organization of the requirements and models design work, the major challenge remains to provide unambiguous references for testing at various stages of the development process and at different Product integration levels. If you are not able to evaluate whether a realization conforms to what stakeholders want and to safety criteria, what's the need for such a costly virtualization? Yet, everyone familiar with operational System Engineering knows it's beyond the reach of existing methods or commercial "solutions", especially when competing viewpoints are at stake. The pragmatic ones focus on a more modest ambition: embedded software code generation.

Calculability just amounts to formalizing a process as a Turing Machine. But up to now, no one has ever come up with a general and physical interpretation of the concepts of "input", of "State" and of "output". Even in mathematics remains a great ambiguity as to the "state" concept. Sometimes it refers to "values" – (space state of a variable), sometimes it refers to a parameter that determines an output according to an input and triggers some "state" transition (Turing Machine).

This ambiguity is symptomatic of the interpretation problem that raises a concept that stands at the interface of the mathematical and physical worlds. For example, usual functional approaches in System Engineering do comply with the calculability criteria. All the same, they have never proved to be efficient in "complex" Systems development processes, despite the amounts invested. A clever look at the situation brings out that this shortcoming is deeply rooted in the functional paradigm[1] .

So, claiming that your System Engineering solution is bound to yield added value in the design of Systems relying overwhelmingly on shared resources, just because it has a mathematical formalism

[1] see http://www.mersyse.com/en/isr_af.php

and that defining a Turing Machine amounts to defining a “System” reveals, at best, a superficial thinking, and a lack of experience, at worst, a fraud for commercial purposes.

On the opposite, striving to set up a scientific and general framework that overcomes these fundamental shortcomings, that aims at actually improving the efficiency of collaborative project or knowledge building processes, lead to a drastic upheaval. It questions the most fundamental concepts that underpin the realistic approach that keeps on dominating implicitly the technical landscape. It imposes to clarify the relationships between human experience, project formalization, description, and Reality. It leads to taking into account, formally and generally, the acting finalities that drive our way of conceptualizing. That’s the demanding approach we have undertaken, with no concession to the scientific criteria of conceptual consistency and refutability, that here stands for operational efficiency: http://www.mersyse.com/en/sr_synthesis.php.