Alternative Views Regarding Digital Transformation and Requirements Engineering in Embedded and Cyberphysical Systems

Carlos Henrique C. Duarte

Digital Transformation (or simply digitization) is about adopting disruptive technologies to increase productivity, value creation, and social welfare [1]. Digitization is perceived as an increasingly important theme by the software engineering community and certainly deserves further discussion with researchers and practitioners.

In a recent paper published in IEEE Software [2], Weyer, Daun and Tenbergen present their views regarding how digital transformation changes requirements engineering in the embedded and cyber-physical domains. They argue that such systems must be functionally safe (e.g., due to correctness) and function safely, and mention cybersecurity threats as possible causes of unsafe behavior. Indeed, correctness is a safety premise, but security and privacy are additional requirements that most embedded and cyber-physical systems need to satisfy in digitization projects.

The authors state that a specification S is a model for a requirement R, given an environment E, formally $E, S \vdash R$. However, S can be regarded as a model for R only if a model-based specification approach is adopted. Algebraic and axiomatic approaches are equally suited to digitization and requirements engineering due to their high abstraction and definitional capabilities. These alternative approaches are often used to capture domain knowledge, elicit customer wishes and needs, and formally represent the most important among them with precision.

Concerning requirements implementation, Weyer, Daun and Tenbergen assert that a program P implements a specification S only if, whenever P is executed on a platform M, P fulfills S. In this case, the system (P, M) satisfies each requirement R supported by S in the context of E, formally $M, P \vdash S$. This formulation does not consider linguistic aspects, but in algebraic or axiomatic settings [3], E and S and R must be written in the same language. On the other hand, P and M can be defined in different languages, and even the symbols in S and P denoting the same real-world dimensions can be distinct. In digitization processes, in which lean or agile methods are frequently used, requirement and software engineers more and more need to work concurrently and continuously together. They have different cultures and accordingly demand the freedom to adopt the idioms that best suit their needs. Neglecting linguistic aspects also hinders reusability, traceability and change management, which are prevalent concerns in requirements engineering.

Regarding the connections between design and implementation, M can be considered a realization of E and P as a realization of S. In case specifications and programs rely on the same linguistic support, we can write $M, P \Vdash R$ for each requirement R

supported by *S.* Still, the development process may be gradual and have intermediate steps [3]. This possibility reminds us to make more explicit and rigorous the respective development transitions, for example, by adopting logical systems [4] connected through formal language mappings [5]. This approach enforces validity, correctness and enables effective computer-aided assistance throughout the process.

The elucidation of development steps and the distinction between specification and program satisfaction is crucial because some specifications are not realizable due to the impossibility to satisfy temporal or reliability constraints [6]. Realizability is a vital context-dependent property in embedded and cyber-physical systems as they aim to solve real problems by relying on hardware or mixed interfaces, e.g., for real-time and feedback control. Realizability is also a relevant concern of requirement engineers due to their productivity, affordability, and time-to-market commitments, particularly in digital transformation projects in which they are expected to deliver market-disrupting solutions.

The context assumptions and functional/quality guarantees of embedded and cyberphysical systems are related to their open and dynamic nature. However, they not always lead to an automated system's composability with its environment, which possibly contains elements with concurrent reactive behavior, such as connected IoT devices. In case we adopt a specification discipline that prescribes dynamic interfaces with a few context assumptions and functional/quality commitments [7], it is possible to ensure the compositionality of designs and the verifiability of emerging progress properties. By strategy, requirement engineers usually formulate broad environment assumptions and minimum viable system obligations to avoid frequent specifications changes.

In real-world digitization projects, change is the rule. That is why Weyer, Daun and Tenbergen go beyond the concerns with openness and dynamic behavior by arguing that embedded and cyberphysical systems should be adaptive. Adaptability is achieved by detecting changes in requirements or assumptions made about the real world, thus reacting accordingly. So, each system must keep a model of its own requirements and environment in this scenario, apart from implementing runtime deductive capabilities, which are also required to support self-learning. We can represent the internalized deductive capability by $\Vdash$ to avoid ambiguity with $\vDash$ and $\vdash$, symbols adopted in the design and implementation stages. Runtime deduction and introspection are important, say, to preserve, adapt or optimize system operation in the presence of changes [5].

Digitization has not inspired radically new software technologies. Instead, it has given rise to new software technology applications, owing to the additional requirements that must be satisfied [1]. In view of the requirements of safety, security and privacy, linguistic and logical rigor, realizability, composability and reasoning, the complex processes observed in the development of embedded and cyberphysical systems are challenging to engineers. Due to the identified technical challenges, requirement engineers' skills and capabilities are possibly more important than the underlying methods, techniques, and tools. The human factor is critical because the required competencies for dealing with high abstraction levels, problem-solving and managing complexity are often lacking or insufficient. This situation highlights the need for knowledge transfer and for improved education, training, and engagement.

Consequently, it is important to salute Weyer, Daun and Tenbergen for presenting their extensive views in [2] so that we were able to contrast them here with our own experiences. Indeed, requirement engineers must balance the needs of change and rigor, particularly in connection to software development. In the multifaceted world embraced by digital transformation, their distinct viewpoints, perspectives, and opinions are most welcome.

References

1. C. Ebert and C. H. C. Duarte. Digital Transformation. *IEEE Software* 35 (4):16-21. Jul. 2018.
2. T. Weyer, M. Daun and B. Tenbergen. The Changing World and the Adapting Machine. *IEEE Software* 38(5):83-91. Sept. 2021.
3. T. Maibaum and W. Turski. On What Exactly is Going on When Software is Developed Step-by-Step. In: *Proc. 7th International Conference on Software Engineering* (ICSE 1984), IEEE Press. pp 525–533.
4. J. Meseguer. General Logics. *Studies in Logic and the Foundations of Mathematics* 129, pp 275-329. Apr. 2000.
5. D. Bouskela and others. Formal Requirements Modelling for Cyber-Physical Systems. *Requirements Engineering*. Aug 2021. doi: 10.1007/s00766-021-00359-z
6. M. Abadi, L. Lamport, L. and P. Wolper. Realizable and Unrealizable Specifications of Reactive Systems. In *Proc. 16th International Colloquium on Automata, Languages and Programming* (ICAPL 1989). vol. 372 *of Lecture Notes in Computer Science*, Springer. pp. 1-17.
7. C. H. C. Duarte and T. Maibaum. A Rely-guarantee Discipline for Open Distributed Systems Design. *Information Processing Letters* 74(1-2):55-63. Apr. 2000.

About the Author

CARLOS HENRIQUE C. DUARTE is a senior technical staff member of the Brazilian Development Bank (BNDES) assigned to work at the Brazilian Institute of Statistics and Geography (IBGE) in Rio de Janeiro, Brazil. Contact him at carlos.duarte@computer.org .