Invited Talk: A Roadmap for Engineering Safe and Secure Cyber-Physical Systems

Alexander Egyed(✉)

Institute for Software Systems Engineering, Johannes Kepler University, Linz, Austria
alexander.egyed@jku.at
http://www.alexander-egyed.com/

Extended Abstract. Safety and Security cannot simply be added to systems. Neither does an architectural choice or design pattern inherently guarantee safety and security. Nor does a safe and secure part of a system make the whole system safe and secure. Ensuring safety and security is an engineering process. This is especially true for Cyber-Physical Systems (CPS) where safety and security concerns transcend hardware and software across different disciplines and across hardware/software subsystems [1].

From an engineering perspective, safety and security reflect functionalities that a given CPS must satisfy. Unfortunately, CPS requirements merely reflect goals that engineers must satisfy without revealing how to satisfy them. The implementation of safety and security concerns is thus a discovery process during engineering – much like how engineering unfolds in general. As engineers define and refine the structure and behavior of CPS – the design – they continuously validate this design structure and behavior against security and safety concerns [6]. For the most part, this implies that:

- Safety and security concerns are discovered incrementally during the engineering process as engineers make design decisions (i.e., changing/augmenting the structure and behavior). This discovery process is reactive and it is not obvious to engineers when they fail to discover a safety or security concern.
- Safety and security concerns are resolved by adapting the design of the CPS. Safety and security concerns thus cause design changes that need to be propagated to all affected engineering disciplines and system/subsystems boundaries [5]. Often, this resolution process is ad hoc and it is not obvious to engineers if they propagated the changes completely and correctly.

In CPS, the discovery and resolution process of safety and security concerns tends to be done separately by every engineering discipline. While not all resolutions affect all these engineering disciplines, many do. Resolving safety and security concerns thus tends to be a multi-disciplinary problem that requires coordinated changes and augmentations to the existing design [2]. This poses a range of challenges to the engineering process.

- Focus on collaboration: for CPS, safety and security concerns may be detectable by individual engineers but their resolution tends to require a coordinated set of changes across different engineering disciplines (co-evolution).

E. H. Abdelwahed et al. (Eds.): MEDI 2018 Workshops, CCIS 929, pp. 113–114, 2018.
https://doi.org/10.1007/978-3-030-02852-7_10

This not only crosses engineering discipline boundaries but also tool boundaries as different disciplines tend to use different kinds of engineering tools [4]. Today, it is not understood how, say, a software change affects the electrical circuitry of a CPS [3].

- Focus on variability: CPS are inherently configurable system – often customizable to specific customer requirements. Here, safety and security concerns transcend variations of CPS. Changes to one variant may affect others [7]. More significantly, we must distinguish how safety and security affect the engineering of a single CPS variant vs. how they restrict a customer from reconfiguring a CPS during runtime – the latter being increasingly vital for self-adaptable, self-healing or self-optimizing systems where customers want increasing control over CPS with unknown effects onto safety and security.
- Focus on modularization: While a safe and secure subsystem of a CPS does not guarantee a safe and secure CPS, a safe and secure CPS cannot be built on unsafe or insecure subsystems. Most companies see modularization as the key to combine software and hardware in smaller, more manageable parts – rather than developing large, monolithic software systems. The safety and security of the system is then the cumulative safety and security of its parts [8]. This relationship is not yet fully understood.

References

1. Biró, M., Mashkoor, A., Sametinger, J., Seker, R.: Software safety and security risk mitigation in cyber-physical systems. IEEE Softw. **35**(1), 24–29 (2018)
2. Clerc, V., Lago, P., van Vliet, H.: The usefulness of architectural knowledge management practices in GSD. In: 2009 Fourth IEEE International Conference on Global Software Engineering, pp. 73–82, July 2009
3. Demuth, A., Kretschmer, R., Egyed, A., Maes, D.: Introducing traceability and consistency checking for change impact analysis across engineering tools in an automation solution company: an experience report. In: 2016 IEEE International Conference on Software Maintenance and Evolution, ICSME 2016, Raleigh, NC, USA, 2–7 October 2016, pp. 529–538 (2016)
4. Demuth, A., Riedl-Ehrenleitner, M., Kretschmer, R., Hehenberger, P., Zeman, K., Egyed, A.: Towards flexible and efficient process and workflow support in enterprise modeling. In: Persson, A., Stirna, J. (eds.) CAiSE 2015. LNBIP, vol. 215, pp. 270–281. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19243-7_26
5. Demuth, A., Riedl-Ehrenleitner, M., Lopez-Herrejon, R.E., Egyed, A.: Co-evolution of metamodels and models through consistent change propagation. J. Syst. Softw. **111**, 281–297 (2016)
6. Egyed, A., Zeman, K., Hehenberger, P., Demuth, A.: Maintaining consistency across engineering artifacts. IEEE Comput. **51**(2), 28–35 (2018)
7. Linsbauer, L., Lopez-Herrejon, R.E., Egyed, A.: Variability extraction and modeling for product variants. Softw. Syst. Model. **16**(4), 1179–1199 (2017)
8. Trubiani, C., Ghabi, A., Egyed, A.: Exploiting traceability uncertainty between software architectural models and extra-functional results. J. Syst. Softw. **125**, 15–34 (2017)