1. Introduction

In the realm of System Engineering and Model-Based System Engineering (MBSE), the selection of modeling tools profoundly influences project success. This discussion compares and contrasts two major tools, UML (Unified Modeling Language) and SysML (Systems Modeling Language), emphasising their differences and implications for System Engineering. It also delves into the conversion of SysML models into executable models, referencing Chabibi et al.'s article (2018) and its relevance to specific system types.

2. Comparison of UML and SysML:

o **Difference 1: Scope and Focus:** UML primarily targets software development and general modeling, lacking specialied constructs for system engineering beyond software. SysML, in contrast, is expressly designed for MBSE, introducing tailored concepts and diagrams for system engineering tasks.

o **Difference 2: Profiles and Extensions:** SysML extends UML as a profile, offering specific extensions and stereotypes for systems modeling, including blocks, requirements, and parametrics.

These distinctions significantly influence tool selection. SysML excels when addressing complex systems with intricate hardware-software interactions, making it a superior choice for holistic system engineering projects.

3. Converting SysML Models into Executable Models:

o UML may suffice for software-focused projects, but SysML's additional constructs shine in complex systems with hardware-software interplay. SysML better supports systems engineering processes.

o SysML's versatility finds particular relevance in aerospace, automotive, and defense industries, ensuring reliability and performance of integrated systems, especially with interconnected subsystems.

Alternative articles by Mohamed et al. (2020) and Song et al. (2020) discuss SysML's capacity to represent models translatable into executable ones, vital for complex systems. Executable models prove most valuable for systems with complex interactions, stringent safety requirements, and real-time constraints, as seen in aerospace and automotive industries.

4. Conclusion

The choice between UML and SysML hinges on project requirements, with SysML preferred for System Engineering due to its specialization. The ability to convert SysML models into executables is especially crucial for complex systems with real-time constraints, ensuring their robustness and reliability. Understanding these tool differences and capabilities is paramount for System Engineering and MBSE practitioners.

Reference

Chabibi, J., Amyot, D., & Suryn, W., 2018. A Method for Converting SysML Models into Executable Petri Nets for Simulation. In Proceedings of the International Conference on Model-Driven Engineering and Software Development (pp. 320-331). Springer.

Mohamed, M.A., Challenger, M. and Kardas, G., 2020. Applications of model-driven engineering in cyber-physical systems: A systematic mapping study. Journal of computer languages, 59, p.100972.

Song, Z., Zhang, Y., & Xie, X., 2020. Model transformation from SysML to executable models for cyber-physical systems. IEEE Access, 8, 20450-20460.