1

Introduction

Model-based system architecture (MBSA) combines the two key technologies model-based and systems architecting. Both are major parts of the future state of systems engineering [123].

Many systems result from a evolutionary development. They are driven by their parts and do not emerge from the architecture. The parts could be anything that, in combination, is assembled to a human-made purposeful system. System architecture is followed by a complete system. Often system architecture is referred to the architecture from the perspective of a software architecture in combination with the hardware or the architecture of software-intensive systems [43]. We understand system architecture more holistic and also consider systems without any software, even though systems without any software, are rarely handled with systems engineering processes and MBSA concepts like described in this book. A system architecture is always present. In today and future systems engineering, it is crucial to apply explicit systems architecting for the success of the system project [123]. Chapter 5 defines the term "system architecture" within its context.

Studies clearly show that systems architecting is critical for the performance and success of the system [68]. This is particularly evident for projects that require significant architectural work or rework. Due to more and more dynamic and complex markets and environments, the system architecture must more and more support the changing requirements and requests for radical changes. Chapter 3 lists the benefits of systems architecting.

A system architecture is about establishing solutions that are in line with the directions that guide the organization and checked for feasibility by the corresponding experts, about designing interfaces that are agreed from both sides, and about ensuring that the people who should know the architecture of a system have a common understanding of it. MBSA uses models for enabling the creation of healthy communication around the architecture of the system and for ensuring that the architecture is validated from different points of view. Models are a key

Model-Based System Architecture, Second Edition. Tim Weilkiens, Jesko G. Lamm, Stephan Roth, and Markus Walker.

tool to be capable of developing complex systems on time and in a feasible quality. Chapter 6 defines the term “model” and MBSA and discusses related terms.

Models are more than graphics. There are even models without any graphical representations. Just the graphics is not modeling but drawing. To create a model, you need the concrete syntax, the abstract syntax, and the semantics, which you find in a modeling language. We use the international standard Systems Modeling Language (SysML) as a language for the system requirements and architecture models. Appendix A gives an overview about SysML, including an outlook on the next-generation modeling language SysML v2. Although we extensively use SysML in this book, our methods and concepts are independent of SysML and could also be implemented by any other modeling language.

The system architect is the one in charge of shaping the system architecture. This is a big responsibility and a big challenge. Organizations developing systems should carefully select people who are allowed to architect the system – and these people’s work results will be tightly monitored by stakeholders everywhere in the organization. Chapter 22 describes how systems architecting could be embedded in an organization, and Chapter 12 discusses the interfaces to the stakeholders of systems architecting. In particular, Chapter 10 introduces the adjacent discipline requirements engineering that closely collaborates with the systems architecting. The SYSMOD zigzag pattern presented in Chapter 9 shows the relationship between requirements and architecture and clearly demonstrates the need for a close collaboration. Artifacts of the model-based requirements and use case analysis are important inputs for the system architects, especially to elaborate a functional architecture using the so-called Functional Architectures for Systems (FAS) method.

Chapter 17 is a comprehensive presentation of the FAS method. Functional architectures are built of functions only and are independent of the physical components that implement the functions. The functional architecture is more stable than a physical architecture that depends on the frequently changing technologies. The architecture principle to separate stable from unstable parts is covered in Chapter 9 about architecture patterns and principles.

Besides the functional architecture, we define and discuss further system architecture kinds. The base architecture that fixes the preset technologies and adjusts the scope for innovation, the logical architecture that specifies the technical concepts and principles, and the product architecture that finally specifies the concrete system. All three architecture kinds are physical architectures. The layered architecture is an additional aspect to these architecture kinds and is presented in Chapter 11.

Another additional aspect is the modeling of variants. Variability is increasingly important. The markets are no longer satisfied by commodity products. The market requests customized products that fit personal demands of the customers.

Additionally, global markets with different local environments and policies require different configurations of a system. Chapter 18 presents a model-based concept to specify different product configurations and gives a brief introduction to model-based product line engineering (MBPLE).

The architecture concepts are presented with a consistent example system. The "Virtual Tour" system (VT system) provides virtual visits by driving with camera-equipped robots through a real exhibition. The system is easy to understand and, at the same time, sufficiently complex to demonstrate the systems architecting concepts. The VT system is also part of a rescue and observation system to illustrate a system of systems and cyber-physical systems. The systems are introduced in Chapter 2.

The system architect who thinks that his job is to make a diagram and save it on a shared network drive will most probably fail. Same for the system architects who think they are the bosses of the development staff and can instruct the other engineers. It is neither an archaeological job nor a chief instructor job. Systems architecting is collaborative work that requires communication and soft skills. A basis for a good communication is a common language and media to transport the information. Chapter 8 covers the artifacts of the architecture documentations. In Chapter 19, we extend our scope to system of systems and architecture frameworks.

Typically, engineers are focused on the technology challenges of their job. As mentioned, communication and more general soft skills are getting more and more important capabilities. The engineering disciplines are growing together. For instance, that could be seen by the modern discipline mechatronic. And the worldwide humankind is growing together due to the internet other communication and transportation technologies. In consequence, an engineer has an increasing number of communication relationships. She is no longer successful when she only manages her technology tasks. It is also important to collaborate well with team members, stakeholders, communities, and so on. Chapter 23 gives an introduction about soft skills for engineers.