Re-using SysML System Architectures

Andreas Korff

Abstract. In the development of complex systems, Model-based Systems Engineering (MBSE) has been introduced successfully into many projects. The OMG SysML as architectural modeling language is part of that success, because it offers the right perspectives to design complex systems. However, even when using MBSE methods and the SysML, the risk of re-inventing the wheel has to be overcome as well. This implies that System Engineers have to know in advance, what is already available as re-usable system asset and fits to the needs of the on-going project.

After introducing a suitable standard with OMG RAS, this paper will show different use cases for component-based design expressed in SysML using asset definition and propagation. This includes top-down exchange of system component specifications as well as bottom-up construction of new systems based on existing components. Asset reuse is based on communication. We will present the necessary communication means for propagating SysML-based system assets, which allows efficient collaboration in teams designing complex systems.

1 Setting the Scene: What Is an Asset?

In times when systems get more and more complex, because their functional and non-functional requirements increase in numbers and interconnectness, in parallel the system design team is dispersed into many sites and organizations, the time available until the system is ready and proven to work is expected to be more and more limited. Currently, there are specific trends visible in a lot of domains, which can help in this situation:

________________________________________________________________________

Andreas Korff
Atego Systems GmbH
Major-Hirst-Str. 11
38442 Wolfsburg
Germany
e-mail: Andreas.korff@atego.com
First, the on-going appreciation of system engineering as a necessary and important role in the organizations. Systems engineers are able to focus on a proper solution design, before going into details too fast. They are bridging between the other domains, project management; hardware, software and mechanical engineering; the regulatory and certification authorities of concern; and last, but definitively not least, customers and project sponsors.

Second, the inclusion of modeling, growing from pure implementation concept into design and specification. Model-based systems engineering is the essential driver for the success of a systems engineer, as he is able to balance all the different perspectives like system specifications, requirements documents, interface control documents, test specifications and so on, which he would have to keep consistent otherwise purely manually.

Third, the growing acceptance of the OMG SysML as to primary standard for a language expressing system models. Of course there are many other, also graphical languages out, but the SysML is capable to integrate them as specific notations for detailing single perspectives. Given a proper tooling support, a SysML model is the central element, the cap stone in the arch between pure textual requirements and implementation artifacts, enabling the bi-directional traceability required by not less than all current development standards like ISO 26262, or DO-178C.

Another aspect to speed up time-to-market time is of course to avoid the famous “Reinventing the wheel”. Although very obvious for everyone, the reality in design projects shows that re-using existing artifacts is harder than it should be. Beside the fact that a lot of times artifact designs are not made to be re-usable – and this is applicable for mechanical, hardware and software artifacts – the biggest problem is to transfer and manage the knowledge what is actually available for re-use.

As a consequence, there is a standard needed defining a meta-model supporting re-usability. In fact, there is a standard available already since 2005: The Object Management Group (OMG), which is responsible for well-known standards like the Unified Modeling Language (UML) or the Systems Modeling Language (SysML), offers in its portfolio of publicly available standard definitions the Re-usable Asset Specification, OMG RAS. Originally focusing purely on software artifacts, the definitions in OMG RAS can be interpreted and extended to describe generic assets. As Figure 1 shows, there are profiles foreseen in the RAS metamodel, for applying the asset definitions to different domains.

For component-based design, which is feasible for both software and systems engineering, there are specific elements necessary for a system asset: First of all, we need ways to store assets according their nature and dependent on the projects they are used in. Therefore Asset Libraries, for the general context, and containers or catalogs have to be available. Containers within libraries allow defining access rights and groups of interest for the team communicating the assets. In addition, optionally projects can be used to group assets together independently from their storage in different containers.