Paper ID: 53950
oral

52nd IAA SYMPOSIUM ON SAFETY, QUALITY AND KNOWLEDGE MANAGEMENT IN SPACE ACTIVITIES (D5)
Quality and safety, a challenge for traditional and new space (1)

Author: Mr. Jan Sommer
German Aerospace Center (DLR), Germany, jan.sommer@dlr.de

Dr. Andreas Gerndt
German Aerospace Center (DLR), Simulation and Software Technology, Germany, Andreas.Gerndt@dlr.de
Mr. Daniel Lüdtke
German Aerospace Center (DLR), Germany, daniel.luedtke@dlr.de

AUTO-CODING DATA TYPE FRAMEWORK FOR THE OSRA USING MODERN C++

Abstract

The model-driven software development paradigm promises to increase the quality of flight software. We propose a new model-based data type framework written in modern C++, which takes auto-coding into account from the beginning. The goal of the framework is to realize type safety as well as value consistency and to provide an intuitive interface to the application developer for defining and working with data types. We use language features introduced with the modern C++ standards to allow extensive validity checks at compile time and additional checks at run-time. At the same time we do not use language features which are generally forbidden in an onboard software context, e.g., dynamic memory allocation or exceptions.

In order to test the feasibility of our code generation framework we use the Onboard Software Reference Architecture (OSRA) as input model for our generator. The European Space Agency developed OSRA with the goal to capture all aspects of modern onboard software, including the definition of data types, in a comprehensive model for later code generation.

We generate the source code from OSRA models in two steps. First, we transform the graphically declared types of OSRA into an ASN.1 textual representation. In return, manually created ASN.1 data type definitions are registered in the OSRA model as external types. The bidirectional integration means that data types, which have been constructed in ASN.1 notation, can also be used inside OSRA models. This helps maintaining more complex data structures in textual form and enables the use of complex data sets from existing projects like TASTE or PATAS to test the feasibility and the limitations of the type system. In the second step the final C++ source code is generated from the ASN.1 data type descriptions.

The proposed data type framework provides auto-generated type definitions and automatically avoids common sources of errors like faulty initialization, out-of-bound access and accidental range overflows by introducing automatic checks. These checks cause compile time errors, if possible, and run-time errors otherwise. In order to provide developers with a practical solution, efforts were made to facilitate integration with existing code bases and third party libraries. We strive to generate complete onboard software projects from the OSRA component model in the future. The data type system provides therefore the basis for that endeavor as it determines the way how the software will exchange data and how the developer will need to interact with them.