Catalogue for System and Software Properties - CatSY

In many sectors such as transportation, space and health, the criticality of the software systems requires a high level of confidence and operational integrity. Formal methods allow early discovery of potential issues which otherwise may be discovered only during the (software) system integration and validation phases. In formal methods, the correctness and validity of design models is expressed in terms of formal properties. Therefore, their proper definition becomes a cornerstone of the early validation. The process of adequate properties specification poses multiple challenges. Requirements, being the main source of the properties to be specified, are often not trivial to be formalised due to the current practice of using natural language specifications. Furthermore, the formal properties may be very complex, which hinders the specification and the use of the formal methods.

In this study, we addressed this problem with an approach based on temporal logics and checks of logical consistency and entailment for the validation of the formal properties. The approach consists of two main steps: 1) Guided formalisation, which defines a set for formal properties capturing the semantics of a set of requirements expressed in natural language; 2) Formal properties validation, which queries the formal specification with a series of checks based on logical consistency and entailment, with the purpose to find issues in the formal properties or to increase the confidence in their correct specification.

The guided formalisation supports the domain expert in the formalisation of the requirements with taxonomy of requirements, which classifies the informal requirements according to standard categories of requirements. The categories and their definition have been derived from ECSS standards and therefore are considered in the specific domain of space. Formal properties are expressed on top of an architectural model, which defines the components of the designed system, their interfaces and their interconnections, at different levels of abstraction. Property patterns and a Catalogue of System and Software Properties (CSSP), classified according to the requirements taxonomy, help the formalisation of requirements without the need to explicitly write the corresponding formal properties.

The formal properties validation is based on checking consistency, i.e. checking if a set of formal properties does not contain a logical contradiction (or in other words if they have a common model), and checking entailment, i.e. checking if a formal property is logically consequence of a set of other formal properties. These checks can be used to check if a set of requirements allows desired or undesired behaviour.

Contract-based refinement is used to verify if the properties of an abstraction level correctly refine the properties of the upper level. In this setting, properties are structured into assumptions and guarantees, and refinement means that while decomposing a component into subcomponent, assumptions on the environment cannot be stronger and guarantees of the implementation cannot be weaker of the upper level.

We extended the COMPASS toolset to support the above methodology. In particular, we defined new AADL property sets for specifying formal properties and contracts associated with the components of the system architecture and for setting the values of the CSSP.

The methodology and the tool support have been evaluated on a case study based on the requirements specification of the Solid State Mass Memory of the BepiColombo mission.