Fault Tolerance View in Event-B Development

Ilya Lopatkin, Alexei Iliasov, Alexander Romanovsky

Newcastle University, CSR, NE1 7RU, UK
ilya.lopatkin@ncl.ac.uk

There is a considerable amount of requirements related to fault tolerance (FT) within any critical system project[1]. We believe that fault tolerance must be formally and explicitly developed starting from the earlier engineering steps with the purpose of improving requirements traceability, development discipline and to allow developers to introduce and evaluate the fault tolerance decisions earlier in the development.

To this end we propose a solution for systematic integration of fault tolerance during the refinement-based formal stage of software development in the Event-B method [3]. A modelling approach is introduced to assist the development of the fault tolerance part of the models. We provide a set of *abstractions* for system modelling from the FT point of view, which can be further refined using basic *templates*. A formal link with Event-B is based on our previous work on modal system modelling [2]. We have developed a combined modelling environment for modal and FT views operating with our abstractions and templates and orthogonal to the existing Event-B model view.

An FT view is a document developed alongside an Event-B model. It describes the design of the fault tolerance features associated with the model in a compact and concise manner. It also offers simple detalisation rules that assist the user in constructing models with a corresponding fault tolerance part. There is a set of rules for formally checking the consistency of an FT view and its Event-B model.

The two basic concepts of the mode view are *mode* and *transition*(Fig. 1). A mode is an abstract description of a system behaviour. A transition always leads to switching from one mode to another. The FT view approach supports two types of transition specialisation and two of mode specialisation. *Error* is a transition leading to a *degraded* or a *recovery* mode. *Recovery transition* leads from a recovery mode back to *normal*. Mode attribution to the specific type is relative and depends on the transition under discussion.

There are certain restrictions to the ways an FT view is structured. For instance, it must contain neither cycles formed entirely from error transitions nor a normal transition out of a recovery mode.

The building blocks of a diagram are primitives describing the initiation of a degraded mode and a transition into a recovery mode. The principle distinction between the two is that recovery is obliged to terminate and pass control back to the mode from which the initiating error originated.

[1] See, for example, our ongoing work within the ICT Deploy project - http://www.deploy-project.eu/

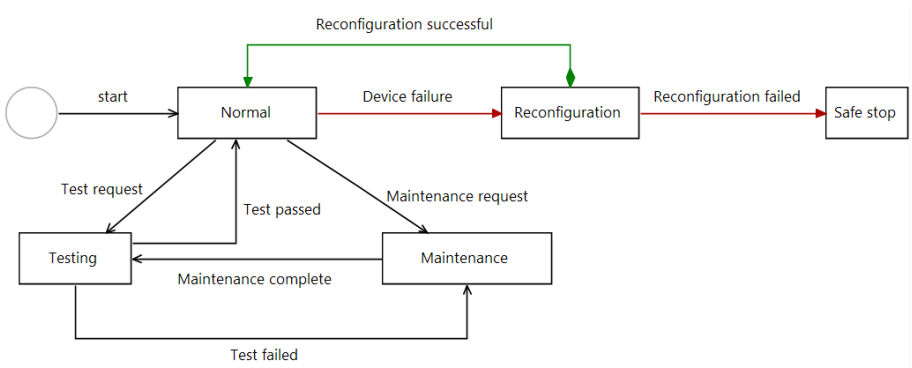

Fig. 1. An FT view example

Diagrams are built in a step-wise manner, starting from the most primitive case and introducing details with a number of predefined templates. At each step, one must show a formal relationship between an FT view and its Event-B model by discharging a number of proof obligations generated by the FT view. *Detalisation* relationship between two consequent FT views is validated by a static checker based on a number of templates available to a user.

Mode is a general characterisation of a system behaviour. To match this notion in terms of Event-B models, modes are mapped into non-overlapping event groups. Likewise, an error is mapped into a single Event-B event.

For a stronger notion of a diagram - model relationship, we consider an FT view as a set of modes providing different functionality under different operating conditions. We use the terms *assumption* to denote the different operating conditions and *guarantee* to denote the functionality ensured by the system under the corresponding assumption. With assumption and guarantee of a mode being predicates expressed on the same variables as an Event-B machine, we are able to impose restrictions on the way modes and errors are mapped into model events and thus cross-check design decisions in either part.

The concept of FT views closely meets the IEEE 1471 standard [1]: it describes the FT "viewpoint" to interested "stakeholders" as a mean to separate the FT "concern". The information about the plugin developed for the Rodin platform will appear soon on the Rodin documentation wiki site.

References

1. ANSI/IEEE Std 1471, also ISO/IEC 42010: Recommended Practice for Architectural Description of Software-intensive Systems.
2. F. L. Dotti, A. Iliasov, L. Ribeiro, and A. Romanovsky. Modal systems: Specification, refinement and realisation. In *ICFEM*, pages 601–619, 2009.
3. I. Lopatkin, A. Iliasov, and A. Romanovsky. On fault tolerance reuse during refinement. In *SERENE 2010*, London, UK, April 2010.