

Modelling Views Paradigm Support for Rodin

A. Iliasov

August 6, 2010

A view is a model part focusing on a specific aspect of a system design [1]. In approach by view we understand a model fragment that cannot be considered in isolation from other views of the same model. A complete model would be obtained by combining all the views. A view differs from a module or a machine contributing to a decomposition refinement since, generally, it is not a complete, self-contained specification and hence does not have to respect the structuring and logical limitations placed on a complete modelling unit.

The notion of view has appeared several times in Rodin tool development and it seems to be a sensible step to come up with common view infrastructure that facilitates the construction of view plug-ins. In our understanding, such infrastructure is also best suited for the introduction of collaborative development. Any contributed view would be automatically compatible with distributed team-based development. One of the main intended uses of the tool is to give modellers an opportunity to use domain-specific notations when using Event B and Rodin while offering tool developers a convenient environment for contributing tools realising domain-specific notations.

The core function of the infrastructure plugin is to manage all the views of a model. It is responsible for obtaining an overall model, running the necessary static checks and generating the proof obligations required to establish views consistency. The primary roles of the views infrastructure plugin are the following:

- filtering; view filters out some model elements and the plugin would automatically merge a view with the rest of a model; also, views of a model may share model elements;
- synchronisation; this includes both the verification of the consistency of logical conditions expressed in differing views and the reconciliation of any shared elements; the latter may require mapping between differing notations if the same element is shared between two views;
- distributed access; the plugin would transparently handle the construction and synchronisation of views of a shared model over network.

The infrastructure plugin does not offer any functionality to a modeller. It is necessary to contribute a plugin defining a views. Such a plugin would provide the definition of a view, an editor (textual or graphical), pretty-printer, static checker and a facility to report proof obligation status. A plugin describing a view does not need to be aware of other views.

One trivial example of a view is an Event B model view. This view would provide an Event B editor conventional in all respects except it would only show

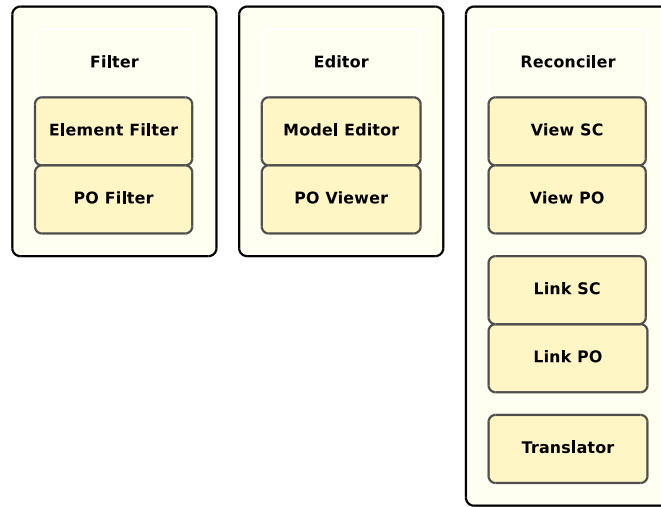


Figure 1: The major logical blocks of the views infrastructure plugin.

a model fragment. One can imagine other views focusing on control flow, data flow, process-based structuring, modal structuring, fault-tolerance, etc.

In the talk, the plugin architecture will be discussed in details along with some methodological implications of the views concept. An example of a view plugin, its structure, interface and typical user session will be presented.

References

- [1] D. Jackson, “Structuring Z specifications with views,” *ACM Trans. Softw. Eng. Methodol.*, vol. 4, no. 4, pp. 365–389, 1995.