# SliceAndMerge: A Rodin Plug-in for Refactoring Refinement Structure of Event-B Machines

Tsutomu Kobayashi[1], Aivar Kripsaar[2], Fuyuki Ishikawa[3] and Shinichi Honiden[1,3]

[1]The University of Tokyo, Japan
[2]RWTH Aachen University, Germany
[3]National Institute of Informatics, Japan

Event-B supports flexible and rigorous modeling with a refinement mechanism based on proof obligation rules. The flexibility enables developers to decide a set of concepts and aspects of target systems focused on in each refinement step. The decision is important in modeling, because it has effects on understandability, maintainability, extensibility of Event-B models.

We focus on refactoring of refinement structure of machines, or re-deciding concepts and aspects of refinement steps in existing proved machines. We aim to realize it in a flexible way through slicing (decomposition) and merging (composition) of refinement steps.

In concrete, when the followings are given:

- Machines $M_A$ and $M_C$ such that $M_C$ refines $M_A$
- $V_B$, which is a subset of $M_C$'s variables

Slicing of the refinement produces an *intermediate machine* $M_B$ such that:

- $M_C$ refines $M_B$ [1] and $M_B$ refines $M_A$
- $M_B$'s set of variables is a superset of $V_B$

Merging is the reverse operation of slicing.

Refinement slicing decomposes introduction of new variables and invariants through a refinement step into several steps. Moreover, it often reveals implicit properties of a concrete machine as explicit expressions of an intermediate machine. Therefore, it helps users to understand descriptions and proof of refinement steps, and thus improves maintainability. A typical problem that can be solved by refinement slicing is a refinement step with a large number of new variables and invariants, which tend to be difficult to understand.

Although Rodin platform supports refactoring expressions of machines and contexts, refactoring of refinement structures has not been tackled.

[1] **refines** clauses of events in $M_C$ are changed through this process.

We implemented a plug-in of Rodin platform named SliceAndMerge to support slicing and merging.

Our approach to slicing is as follows:

1. Calculate variables in the intermediate machine considering dependencies of variables and invariants.
2. Find fragments of $M_A$ and $M_C$ that should also be specified in $M_B$.
3. Add complementary specifications to the fragments, so that $M_B$ becomes consistent.

SliceAndMerge supports users by automating step 1 and step 2 of the above. Users of SliceAndMerge can select a part of invariants of an intermediate machine $M_B$ from a list of invariants of a concrete machine $M_C$ (Fig. 1). Then the tool resolves dependencies between invariants and variables, finds fragments of $M_A$ and $M_C$ that should be included in the intermediate machine $M_B$, and generates a part of $M_B$. The tool also supports merging of refinement steps.

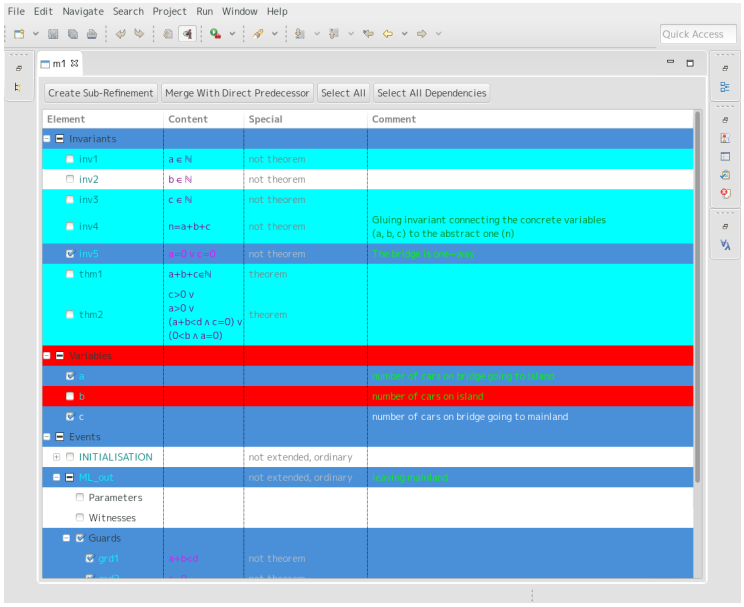


Figure 1: Interface for selection of invariants in intermediate machine

In this tool development presentation, we describe details of SliceAndMerge and give a demonstration of the tool.