A State-based Refinement Technique for Event-B

ISP RAS
Alexey Khoroshilov, Victor Kuliamin,
Alexander K. Petrenko, Ilya Shchepetkov
shchepetkov@ispras.ru

Oryol, September 25, 2020

Formal Models

Formal model is a **rigorous** mathematical specification of a target system at a certain **abstraction** level, which usually consists of:

- state of the system
- properties of the system formulated as requirements on its state
- the behavior of the system in the form of operations that transition the system from one state to another

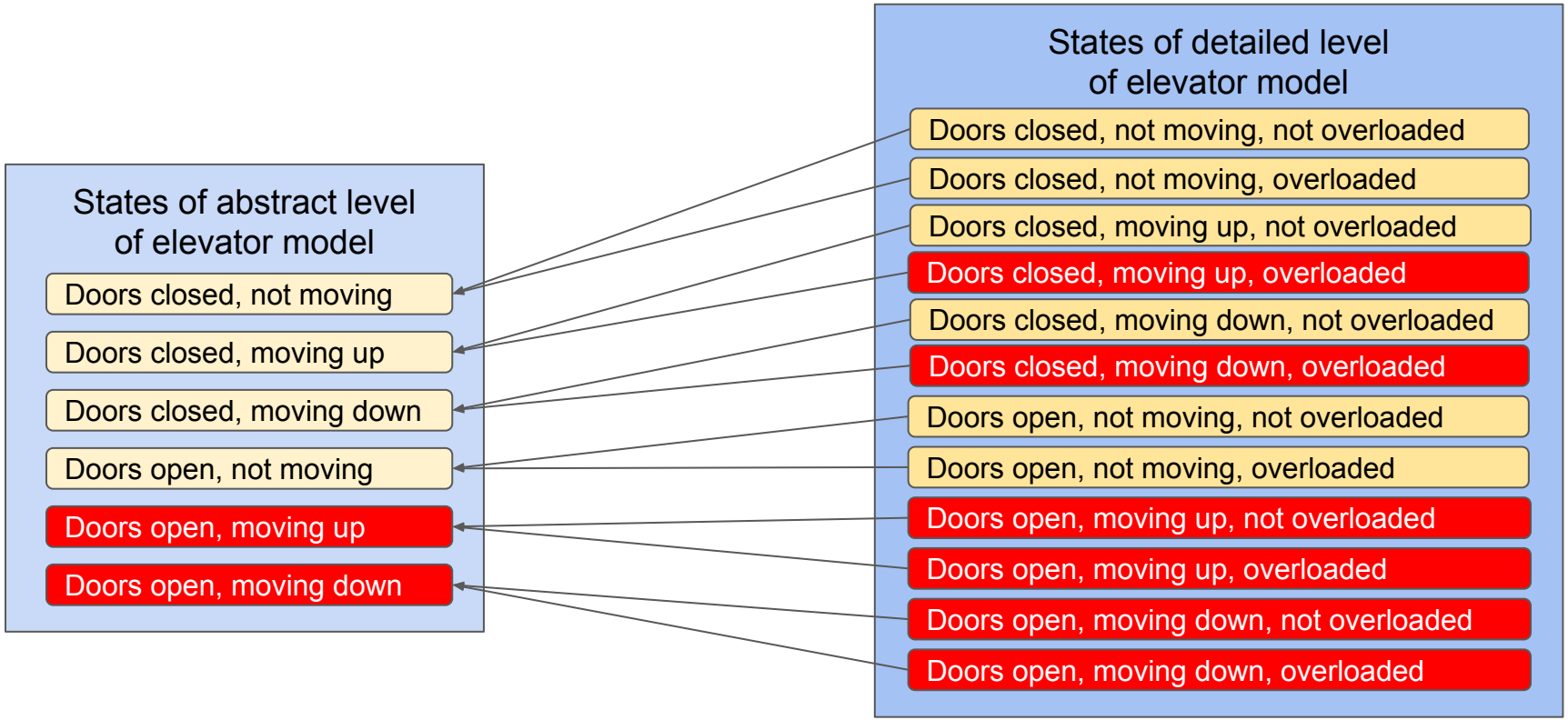
Formal models are used to find subtle errors or to formally prove the absence of errors in the safety-critical systems

Stepwise Refinement

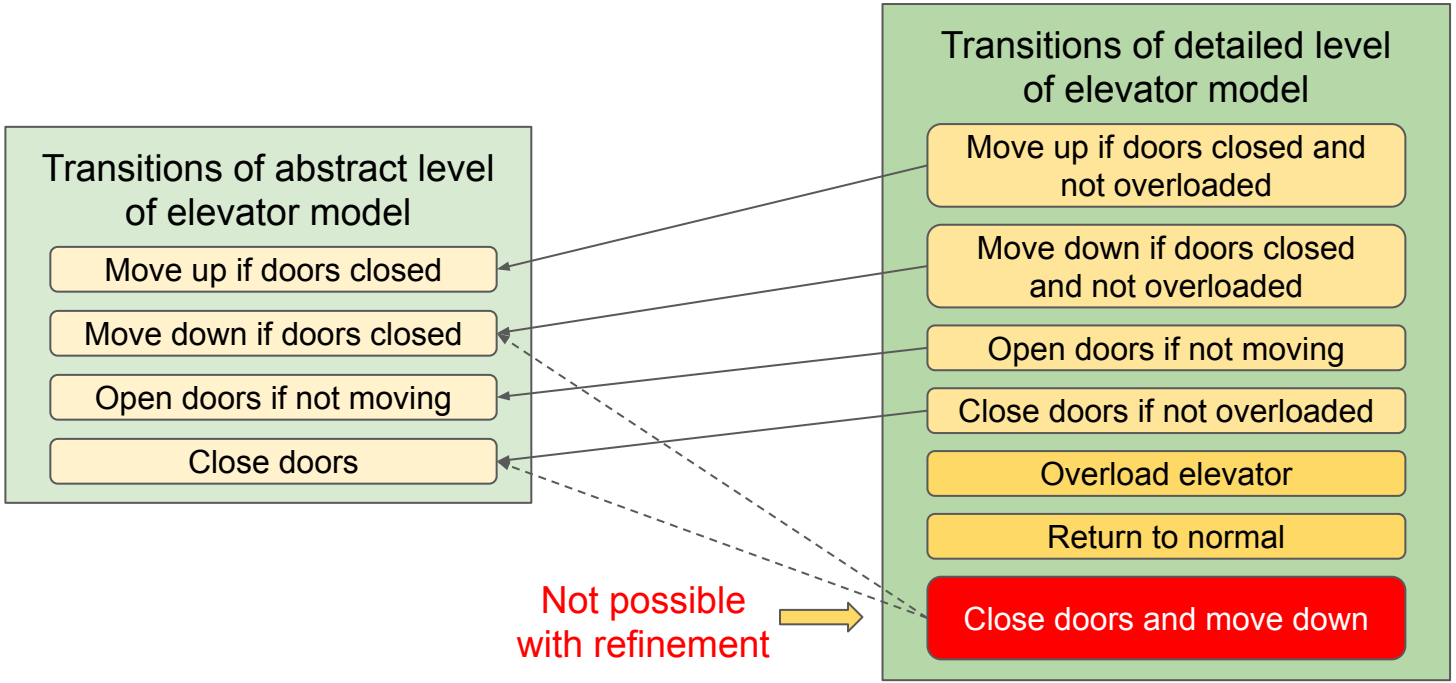
Refinement is a technique of developing models by starting with the simple, abstract model of the system, and then incrementally refining it adding more details at each subsequent level of refinement

Refinement is used mainly to simplify development and verification, but it can be also used to represent systems at different abstraction levels

Stepwise Refinement: States



Stepwise Refinement: Transitions

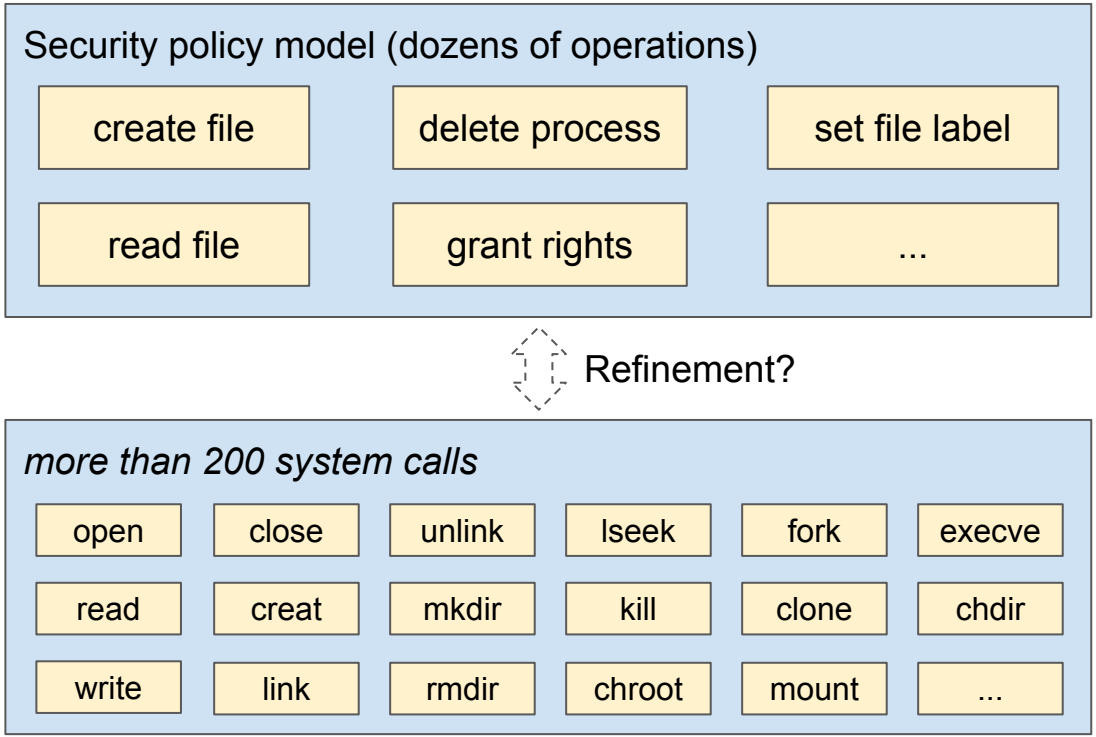


Our Projects

We use formal method Event-B to formalize and verify **security policy models** of various **special-purpose operating systems**. This usually consists of the following steps:

- We formalize and verify the security policy model itself
- We develop an additional, more detailed **formal model** of the system call interface of the OS kernel
- We prove that there is a **correspondence** between the security policy model and this additional formal model: **this is done using the refinement technique**

Issue with Refinement



Goal

Our goal is to develop a new refinement technique for Event-B that can be used to facilitate the development of formal models when drastically different levels of abstraction are needed to be combined

Event-B

Event-B is a formal method for system-level modelling and analysis. Features:

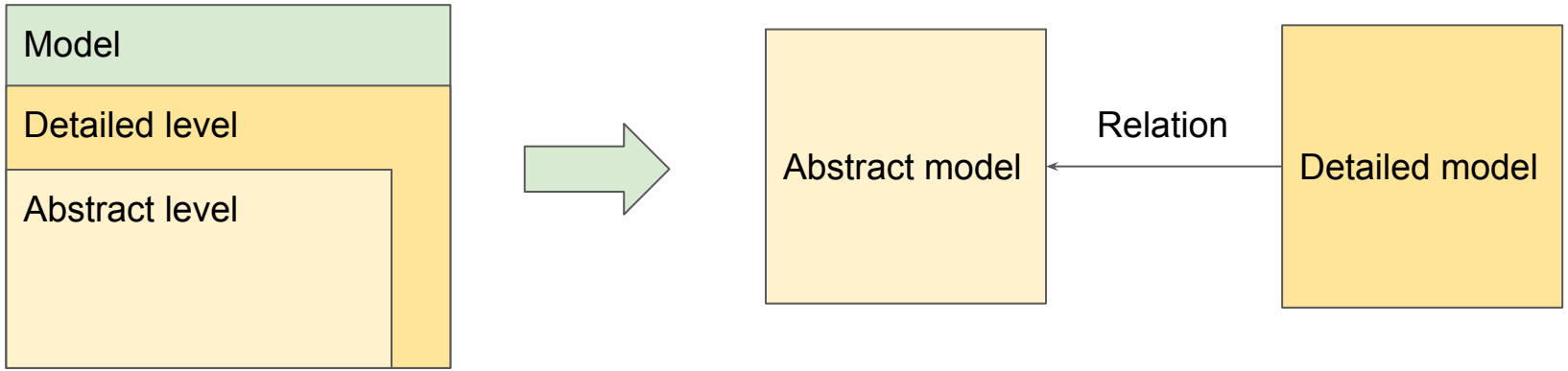
- Easy to learn, based on set theory and first-order logic
- Refinement support to represent systems at different abstraction levels
- Good instrumental support (Rodin platform)
- Ability to use automatic and interactive provers, model checking to verify consistency and correctness of the models

Event-B models are discrete transition systems and consist of:

- Variables (state)
- Events (change state, transitions)
- Invariants (constrain variables, represent requirements)

State-based Refinement

We suggest splitting such "incompatible" levels into separate formal models, which we call **abstract** and **detailed**, and establishing the correspondence **relation** between their **states**



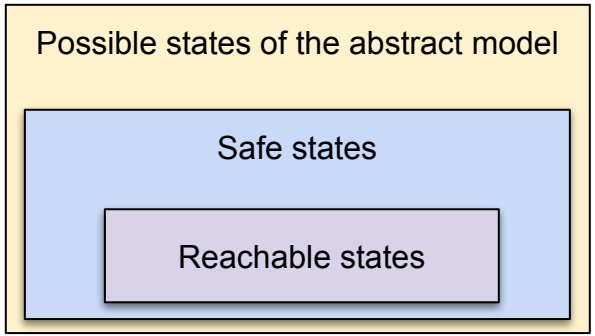
State-based Refinement

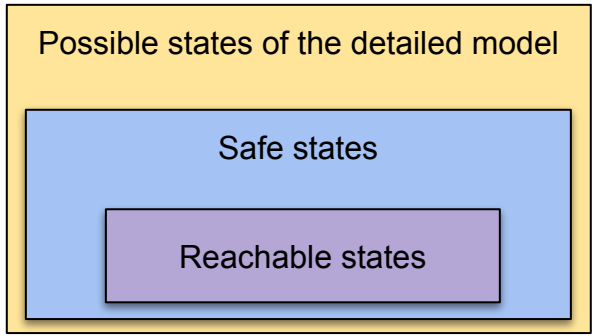
Conventional refinement allows **reusing** invariants and proofs: once invariant is established to be true on one refinement level, it is guaranteed to be true on all the following levels

With the suggested technique, there are no explicit connections between separate models, so such important invariants need to be replicated in each one of them

State-based Refinement

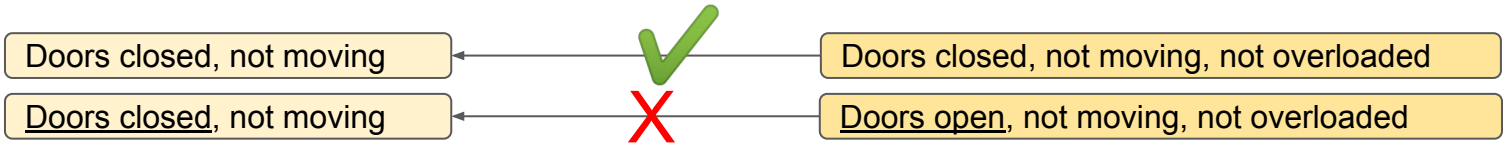
Events of each model are not restricted at all: they can describe the behavior of the system completely differently. These events, however, need to preserve all established invariants of the model they belong to



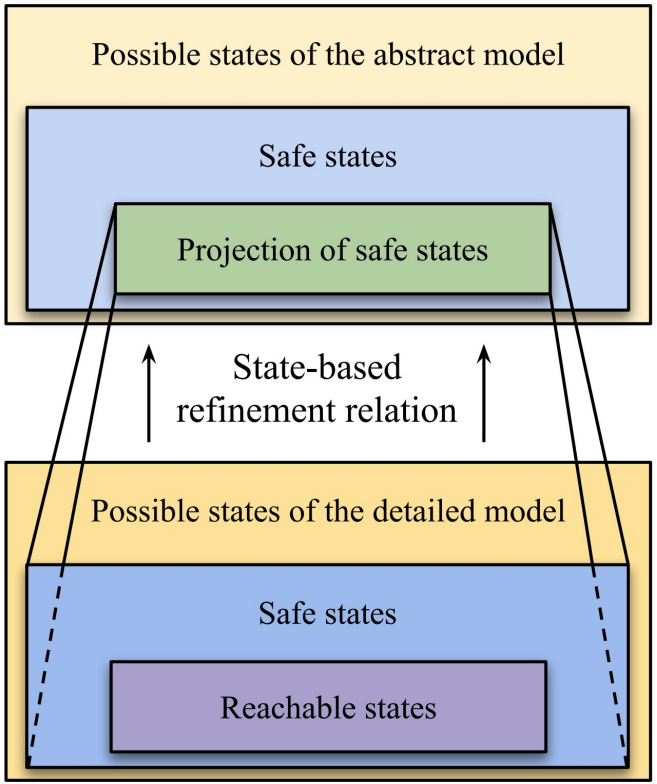


State-based Refinement

States of abstract and detailed models should correspond to each other since they describe the same system:



State-based Refinement



Implementation in Rodin

There are two main features:

- The technique can be used together with Event-B and Rodin without any modifications of existing language and framework
- The correctness of separate models and established state-based refinement relation between them can be proved using automated verification tools (but some manual steps are involved)

Practical Evaluation

We had previous (negative) experience using conventional refinement with security policy models. This time we have introduced the state-based refinement technique and formalized the security policy model and system call interface as separate Event-B models

Though we couldn't reuse some proofs, which is possible with conventional refinement, and had to repeat the security properties in both models, overall the suggested technique turned out to be a success for us and allowed us to achieve the outlined goals

Conclusion

- Formal models are developed with refinement, but its rules are often too restrictive
- We have presented a new state-based refinement technique for Event-B
- This technique allows to sidestep existing refinement rules and develop models which are hard or impossible to develop using conventional refinement
- The correctness of separate models and established state-based refinement relation between them can be proved using automated verification tools

The research was carried out with funding from
the Ministry of Science and Higher Education of the Russian Federation
(the project unique identifier is RFMEFI60719X0295)

Thank you!
Questions?

The research was carried out with funding from
the Ministry of Science and Higher Education of the Russian Federation
(the project unique identifier is RFMEFI60719X0295)