Context instantiation plug-in: a new approach to genericity in Rodin*

Guillaume Verdier[1], Laurent Voisin[2]

[1] LMF/CentraleSupélec, Université Paris-Saclay
guillaume.verdier@irit.fr
[2] Systerel
laurent.voisin@systerel.fr

1 Introduction

The Rodin platform [1] offers an integrated development environment for designing software with Event-B [2]. One of the limitations of its core language is the lack of genericity: it is not possible to define generic data structures or to prove abstract theorems that hold for any type. When such generic constructs are needed, users end up copying and pasting large parts of contexts just to change a type. This approach is obviously cumbersome and error-prone. Alternatively, some plug-ins offer facilities for type parametricity. In particular, the Theory plug-in [4] provides many extensions to the mathematical language that can be used in Rodin, one of them enabling type parametricity. However, these extensions are introduced through a new type of files, *theories*, requiring users to learn how to write these theories and to go through a process of deployment to make them available in Rodin contexts.

As part of the EBRP project, a new, lighter approach has been proposed by Jean-Raymond Abrial. It is based on existing contexts and simply adds an option to instantiate theorems of a context in another one. Intuitively, it is similar to the copy-and-paste method, but done in a safe way by a plug-in rather than manually.

2 Plug-in usage

Defining a generic context can be done using just core Rodin tools, without the plug-in itself. Any context can be instantiated; its carrier sets and constants can be used as generic parameters and substituted with a concrete type or value during instantiation.

Instantiating theorems of a generic context is done by creating an axiom and specifying what it instantiates in the comment box. This specification identifies the instantiated theorem (its name, the name of the context in which it is defined and optionally the name of the project) and may provide some type or value substitutions. It can be manually written or generated through a graphical wizard.

In its most basic operating mode, the plug-in checks that the predicate of the axiom matches the predicate of the instantiated theorem, after applying the substitutions if necessary. Therefore,

*This work is supported by the French ANR project Event-B Rodin Plus (EBRP, ANR-19-CE25-0010).

Rodin users who used to manually copy-and-paste generic theorems can just add comments describing the instantiation and the plug-in will check that no errors were made.

For new developments, users can leave the predicate empty and merely provide the specification of the instantiation: the plug-in can generate the predicate of the instantiated theorem.

Since the plug-in uses Rodin's contexts, users of the plug-in can freely share their work with other users who do not have the plug-in installed: they will see normal contexts with comments describing the instantiations and will be able to use them as any other context.

3 Future improvements

This new plug-in is still in an early stage of development and could use various improvements. The features highlighted in this document are therefore subject to change.

Instantiation is currently done on a theorem basis, which can be cumbersome if one wants to instantiate many theorems from a single context, although the wizard can generate all the instantiations at once. It could be more user-friendly to instantiate contexts first, similarly to the import mechanism, and then be able to use their theorems freely.

When a theorem is instantiated, an axiom is created: considering that the theorem has been proved on some abstract types and constants, there is no need to ask users to reprove it on more specific types or values. However, some issues appear if the proof of the instantiated theorem depends on axioms that may not hold on the substituted types or values [3]. Different solutions have been proposed, such as imposing some restrictions on axioms in instantiated contexts or adding these axioms as proof obligations of the instantiation.

4 Conclusion

The context instantiation plug-in offers a new approach to genericity in the Rodin platform directly integrated in contexts. It should be easy to adopt in existing projects and require little changes to be useful. It is already being used by members of the EBRP project. In particular, Jean-Raymond Abrial and Dominique Cansell have implemented an extensive set of generic contexts ranging from generic data types (such as lists and binary trees) to mathematical theories.

The plug-in is still under active development and should be publicly released soon.

References

[1] Jean-Raymond Abrial, Michael Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. *Rodin: an open toolset for modelling and reasoning in Event-B.* International Journal on Software Tools for Technology Transfer, 12(6):447–466, Nov 2010.

[2] Jean-Raymond Abrial. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[3] Jean-Paul Bodeveix and Mamoun Filali. *Event-B formalization of Event-B contexts.* Submitted.

[4] Thai Son Hoang, Laurent Voisin, Asieh Salehi, Michael J. Butler, Toby Wilkinson, and Nicolas Beauger. *Theory plug-in for Rodin 3.x.* CoRR, abs/1701.08625, 2017.