Using Rodin with Projects on github

Systerel, France

Abstract

This document explains how Rodin projects on github can be imported into the Rodin platform. This is illustrated using the `model-evaluation` sub-project of `openETCS`.

1 Prerequisites

This section shortly describes the basic prerequisites to use Rodin projects on github. First the installation of the Rodin platform itself, then the basic plugins required to use the provided models and finally additional plugins which facilitate usage and extension of the provided models.

1.1 Rodin Platform

This illustration uses the Rodin platform 2.7 [1], for general information on Event-B, Rodin, various plugins etc. see the Event-B Wiki [2]. More details on the installation on Rodin can be found at `http://www.event-b.org/install.html`.

1.2 Basic Plugins

For plugin installations, it is recommended to tick the "Contact all update sites during install to fin required software" (see Figure 1). This will install all necessary dependencies for each plugin, in case these are not yet installed.
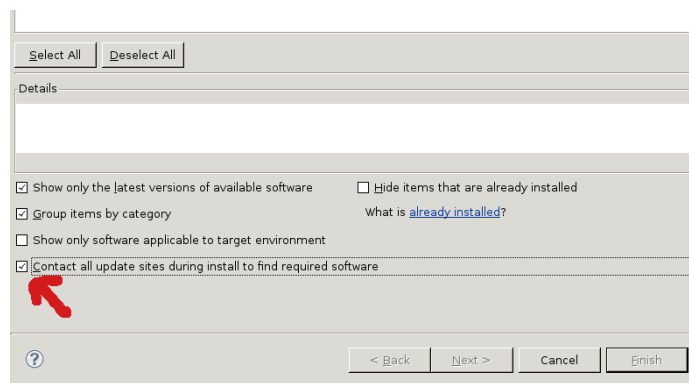

Figure 1: Plugin Installation

---

[1]`http://www.event-b.org/`
[2]`http://wiki.event-b.org/index.php/Main_Page`

Atelier-B Provers The Atelier-B provers facilitate the construction of the proof trees for Rodin proof obligations, by automatically discharging may proof obligations. Their repository is available under Rodin as "Atelier B Provers".

ProR Integration The integration with ProR allows for the traceability of requirements in the provided ReqIf files. The repository is available under Rodin as "ProR", where the "ProR Rodin integration feature" must be installed. **NOTE:** For the current ProR version 0.6.0 there are known problems with Java version 6. It is suggested to use Java version 7, if possible the latest JVM from `http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html`.

EGit The integration with github is done via the EGit plugin. This plugin allows to collaborate on Rodin models and to push/pull changes to github. The repository is available under the official Eclipse repositories as "Indigo Update Site" where the "Eclipse EGit" must be installed.

1.3 Additional Plugins

The additional plugins are not strictly necessary to analyze and inspect the model. Nevertheless, it is suggested to install them if an extension of the model is intended.

ProB The ProB plugin provides means for model-checking and animation of Event-B models. This requires finite instantiation of carrier sets and selection of an initial state. In this situation, the plugin can verify deadlock freeness and LTL formulas or can animate a system run. It is available from the ProB repository under "ProB for Rodin 2".

SMT Solver Plugin The SMT solver plugin will in general lead to a higher degree of automation for the formal proofs. Experiments with industrial cases studies reduced the number of non-automatically discharged proof obligations to one fourth. The plugin comes bundled with two different solvers (CVC3 and veriT) and it can be extended with various other, e.g., z3 from Microsoft or MathSAT5 from FBK. It is available from the Rodin repository under "SMT Solvers Integration".

iUML-B State Machines The "iUML-B State-Machines" plugin allows for graphical modeling of state-machines in an Event-B model. The state machines can be transformed into Event-B code. It has a good integration with the ProB plugin which allows for graphical animation of the machines.

Project Diagram The project diagram plugin allows for visualization of the structure of a model. In particular it visualizes the refinement relations between the machines, the extension relation between the contexts and the sees relation between machines and contexts. It is available from the Rodin repository under "Event-B Project Diagram Plugin".

2 Importing Rodin Projects from github

After the installation of Rodin and the necessary plugins, projects can be imported from github into the local Rodin workspace. In the following, this is explained using the Eclipse project creation wizard.

The first step is to select "File → New → Event-B Project" from the Eclipse menu and then to select "Git → Projects from Git" as import source (see Figure 2).
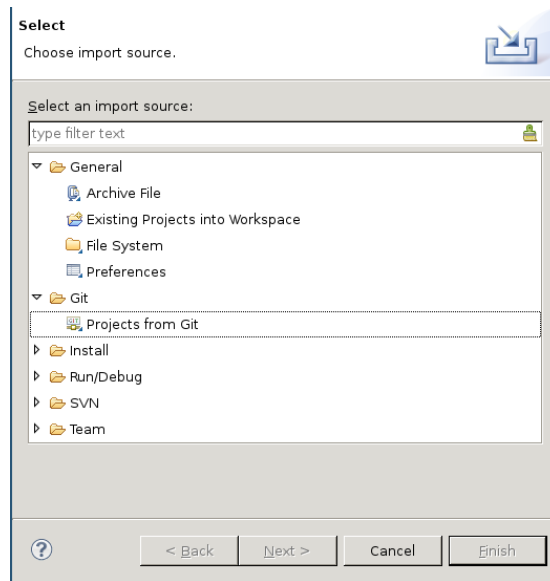

Figure 2: Creating a New Project

The next step is to specify where to find the git repository. For this, one has to select the URI option as shown in Figure 3.
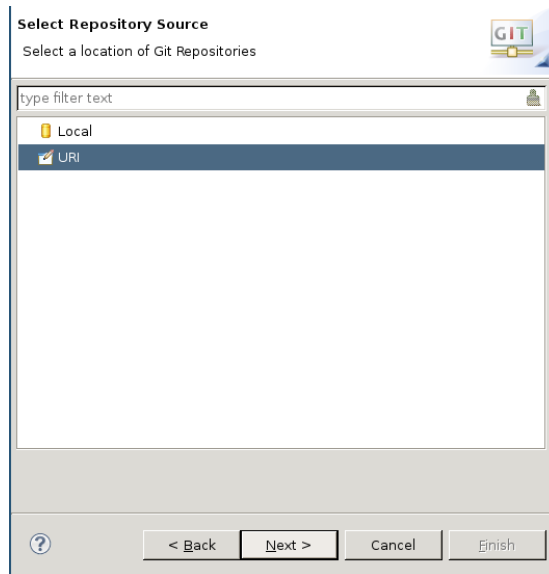
Figure 3: Selection of Repository Source

The next step is to specify the URI of the repository on github. For the model evaluation project, this is `https://github.com/openETCS/model-evaluation.git`. This step also requires the specification of the authentication data, i.e., the username and password for github (see Figure 4).
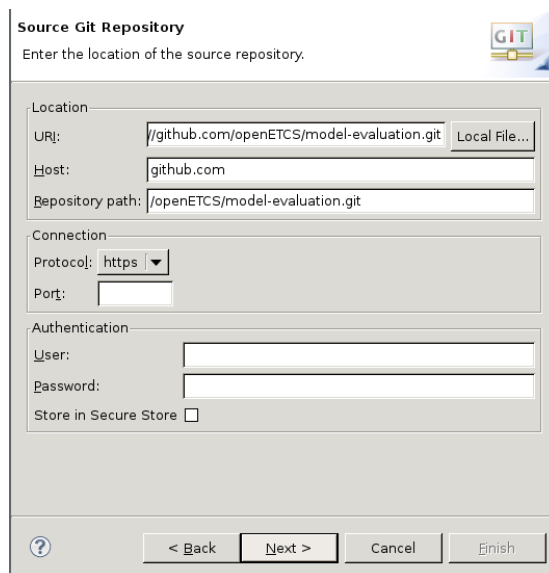
Figure 4: Specify github Repository

The next step is to select the desired branch of the repository. If in doubt and there is more than one branch, selecting the "master" branch, as shown in Figure 5 should be ok.
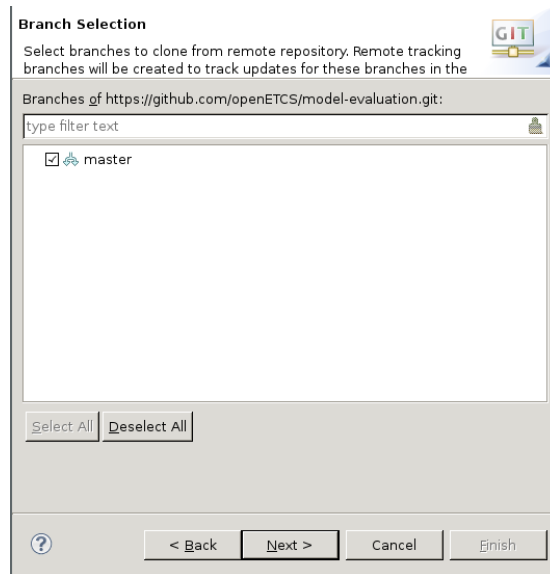
Figure 5: Select Branch of Repository

For a local copy of the repository on github, an empty directory on the local machine must be selected (or created) and a name of the remote repository can be specified, the default is "origin" (see Figure 6).
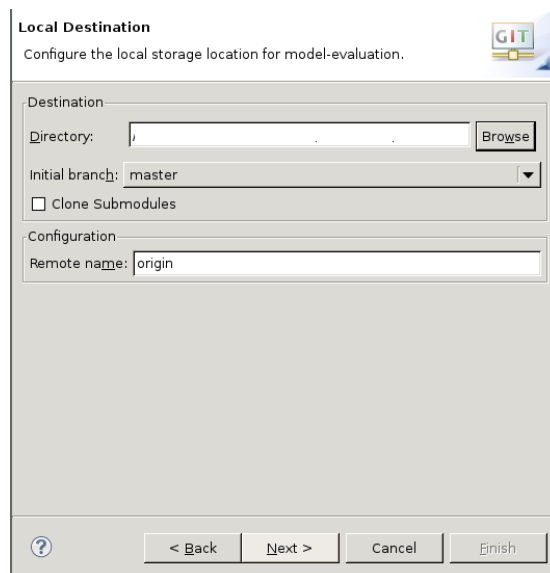
Figure 6: Local Repository Copy

As there can be multiple projects in a repository, e.g., from different tools as for the model-evaluation project, the correct one must be selected. To achieve

this, the collapsed directory tree as shown in Figure 7 must be expanded[3].
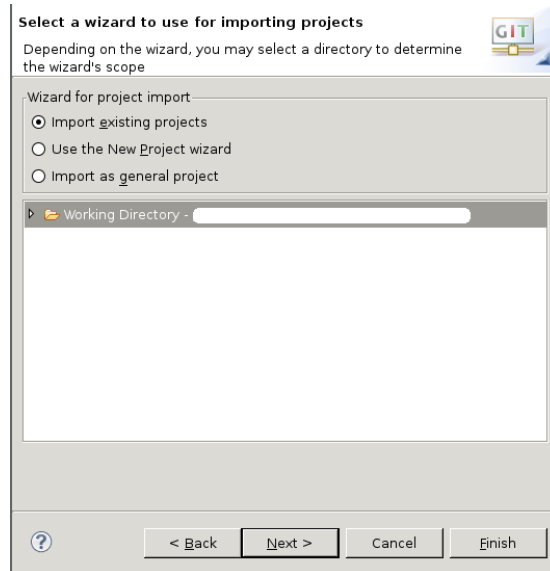
Figure 7: Select Project to Import

In the expanded tree, the project root directory of the project must be selected as shown in Figure 8.
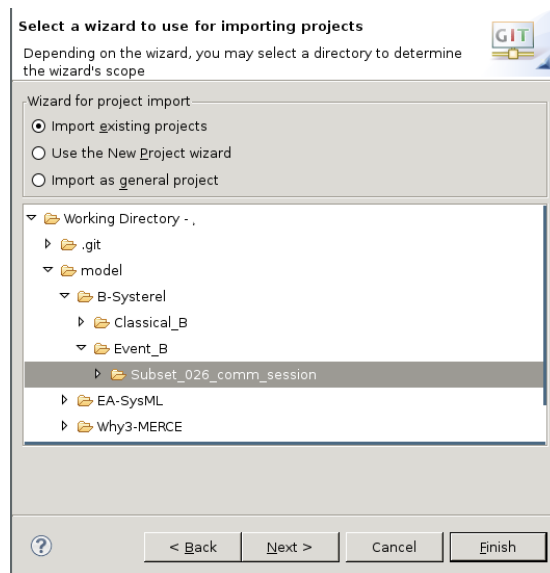
Figure 8: Expanded Directory Tree

---

[3]Sometimes the information which directories are available is not shown directly (the collapse / expand symbol is lacking besides the "working directory"). In this case it helps to push the "back" button to the previous window and return with "next".

Once the correct directory has been selected, the contained projects can be selected as shown in Figure 9 and imported into the local Rodin workspace.
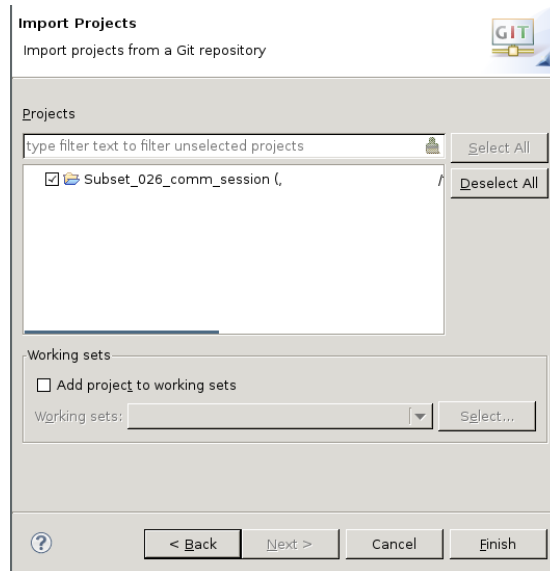
Figure 9: Import Project into Workspace

3   Frequently Asked Questions - FAQ

- **How are requirements traced?**

  For requirements tracing we use the ProR plugin [4] from formalmind [5]. It is based on the OMG standardized ReqIf format. It allows for tracing of Event-B modeling artifacts in ReqIf files. Changes in either the model or the ReqIf are marked automatically, the ProR view of Rodin is shown in Figure 10.

---

[4]http://wiki.event-b.org/index.php/ProR
[5]http://www.formalmind.com

Figure 10: Requirements Tracing with ProR

To open a ReqIf file, the "Resources" view of Rodin must be selected which will show all files in the project in the Project-Explorer.

- **Proof obligations are not shown in the Event-b explorer. How to create them?**

This can have different causes. Trivial proof obligations are not shown, this concerns in particular proofs for well-definedness [6]. It is also possible to hide proven proof obligations by toggling the green button on top of the Event-B Explorer shown in Figure 11 (button is inactive in this example).

---

[6]see http://handbook.event-b.org/current/html/well_definedness_proof_obligations.html
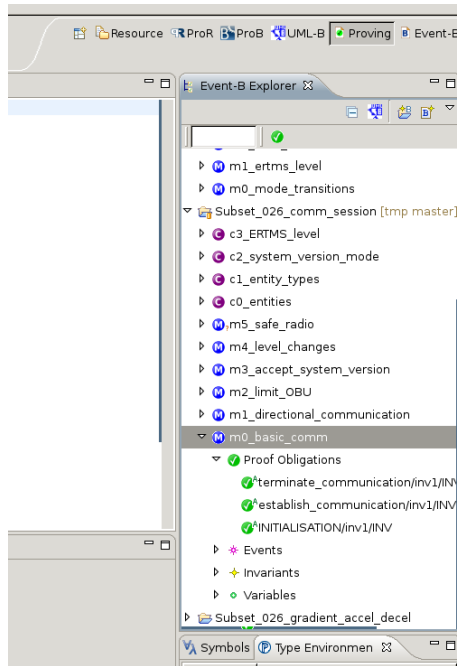
Figure 11: Event-B Project Tree View with Proof Obligations

Another possibility is a problem with some installed plugins, in this case the project can be cleaned by selecting "Project-¿Clean" from the main Rodin menu. If "Project-¿Build Automatically" is selected the project will be automatically refreshed, this is also possible manually by selecting the project, hitting F5 or selecting "Refresh" from the right-click menu.

- **If I try to open a ReqIf file, Rodin nothing happens and Rodin freezes. What happened?**

  There is a problem in the ProR plugin which can cause this behavior (e.g., see here `https://bugs.eclipse.org/bugs/show_bug.cgi?id=397672`). Currently the best solution is to install the most recent JDK from `http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html`.

9