Rodin 软件用户使用手册

二零一四年十一月二十日

目录

1 引言

Rodin工具支持对Event-B形式化方法的应用，它对Event-B模型语法分析与基于证明的验证提供核心的功能。B方法是由Jean-Raymond Abrial 在20世纪90年代初开发的，主要是支持软件的形式化开发，Event-B不仅仅只对软件组件进行建模，而且也对系统进行建模。Event-B主要是对系统进行建模和论证的，这些系统包含了许多物理组件，电子器件和软件。

Rodin工具主要由Jean-Raymond Abrial 开发而成，现在最新版本为3.0版本，能运行在微软Windows和苹果mac机器上。

手册说明

本使用手册，向用户介绍了 Rodin 的运行环境、软件功能、软件安装以及使用方法等。

参考资料

Event-B 主页：http://www.event-b.org

Rodin 工具主页：http://www.event-b.org

Rodin 工具下载链接：http://www.event-b.org/install.html

2 简介

Event-B 模型由两个基础部分组成：机器(machines)和环境(contexts)。机器包含模型的动态部分，即变量(variables)、不变式(invariants)、定理 (theorems)、变体(variants)和事件(events)。环境包含模型的静态部分，即载体集合(carrier sets)、常量(constantsX 公理(axioms)和定理(theorems)。

环境适用于常量,它可以将形式化模型中的参数隔离出来。机器用来封装一个状态变迁系统,其状态用变量集合表示,状态的变迁由带卫兵事件表示。

机器和环境有以下关系:一个机器可以由另一个机器精化(refined)得到,而一个环境可以由另一个环境扩展(extended)得到。并且,一个机器可以看到 (see)一个或若干个环境。

大体来说,Event-B 机器中定义的数学模型是通过状态(State),以及相应的状态变迁(Transition),也即事件(Event)来表示的。状态被定义为变量以及变量所必须满足的某些恒定性质:这样的性质被称为状态的不变式。每一个状态变迁由两个项来定义:变迁卫兵(Guard),定义了状态变迁时所必须要满足的条件,以及变迁行为(Action),定义了状态变迁对当前状态的修改。可以看出,任意的“状态机(State Machine)"都可以用这样的方式来定义。我们需要通过证明的方式来保证状态变迁仍然能够保持状态变量的性质(不变式)。

3 运行环境

3.1 硬件要求

版本为 2.8 的 Rodin 工具大小大概在 130MB，运行 Rodin 工具对 CPU，内存大小，显示器等硬件要求的并不是太高。它是个极易使用的建模与论证工具。

3.2 支持的操作系统

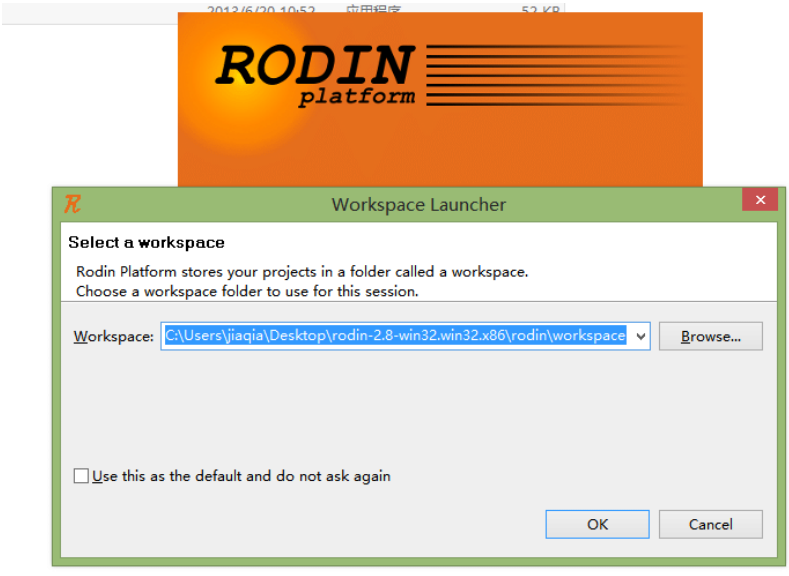
Rodin工具主要由Jean-Raymond Abrial 开发而成，现在最新版本为3.0版本，能运行在微软Windows和苹果mac机器上。在官网上下载下来就可以直接使用,不用再次编译。

4 安装

注意：Rodin 工具有 Windows 版本的也有 mac 版本的，但是在下载时不用刻意去寻找某个版本，如果你当前的系统是 Windows 系统，用 IE 或其他可用浏览器打开下载链接，直接下载，下载下来的就是 Windows 版的，同样在 mac 系统下，直接下载就是 mac 版的，不用刻意去寻找。Mac 系统中下载界面如下：

4.1 Windows

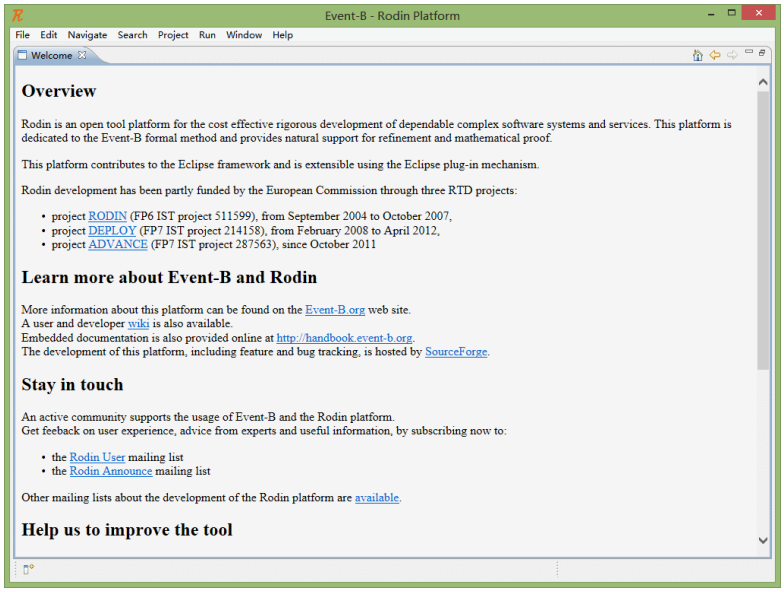
下载 Windows 版本的安装文件。Windows 版的 Rodin 工具不需要安装，把 Rodin 工具文件下载下来后，直接运行 rodin.exe 文件就可以直接运行了。如图一。



图一 开始运行界面（一）

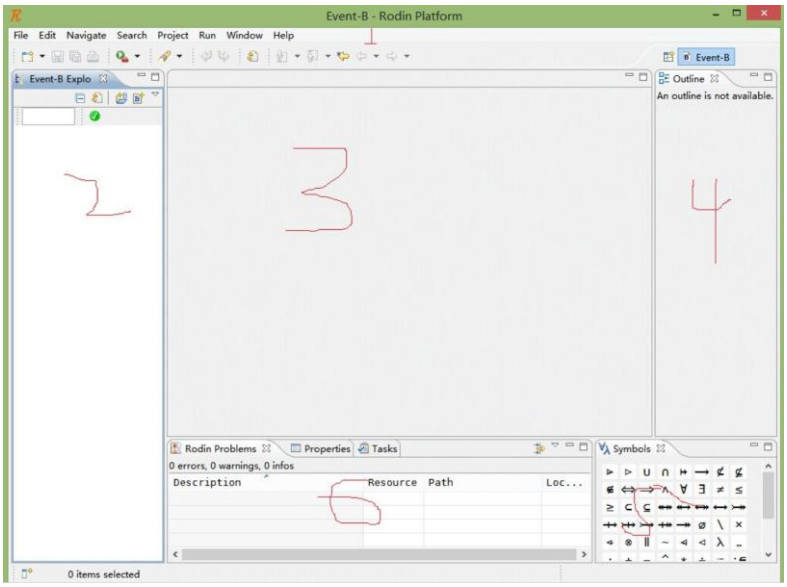
如图所示，首先可以选择工作地点，就是将工程文件放置在哪里的地方。然后点击 OK 按钮，就可以进入运行界面了，可以使用 Event-B 进行建模论证了。

第一次打开的界面是这样的：



图二 第一次运行界面（二）

点 Welcome 旁边的差号就可以进入工作界面，工作界面如下图所示：



图三 工作界面（三）

如工作界面所示，1 是菜单栏和工具栏，2 是 Event-B 工程目录，3 是编写工作界面，4 是文件大纲，5 是一些可用的数学符号，6 是 rodin 问题和 event-b 属性栏。

5 常用操作

现在通过例子来具体讲述 Rodin 工具的用法和一些 Event-B 的知识。

5.1 例子—热水器建模验证

首先，创建个Event-B项目File >New > Event-B Project.

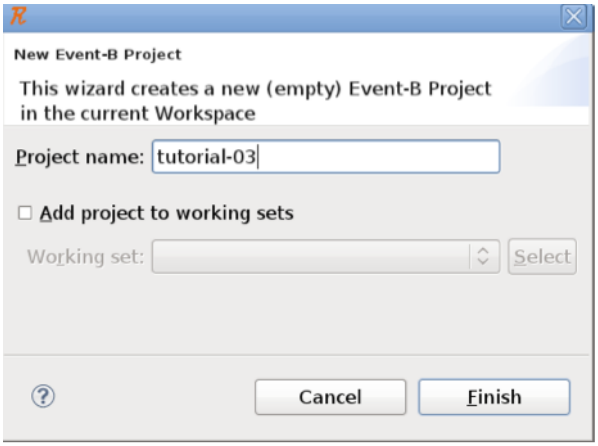


图5.1 创建新的Event-B工程

接着就是创建一个Event-B的Machine组件，File>New > Event-B Component

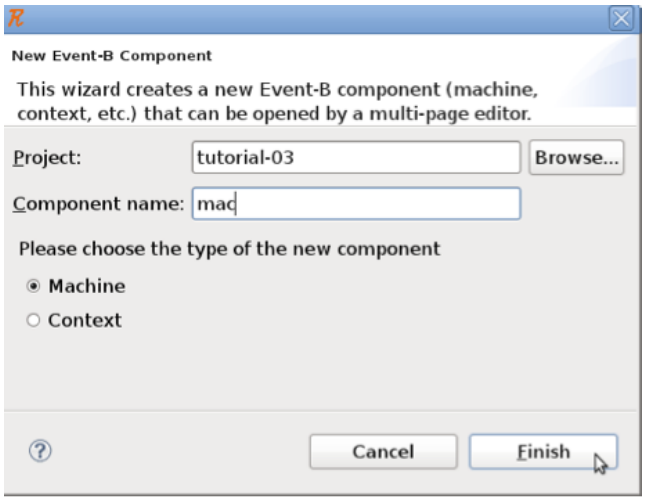


图5.2 创建Event-B的Machine组件

然后就可以在machine中进行操作了，可以创建不变式，时间，变量等。

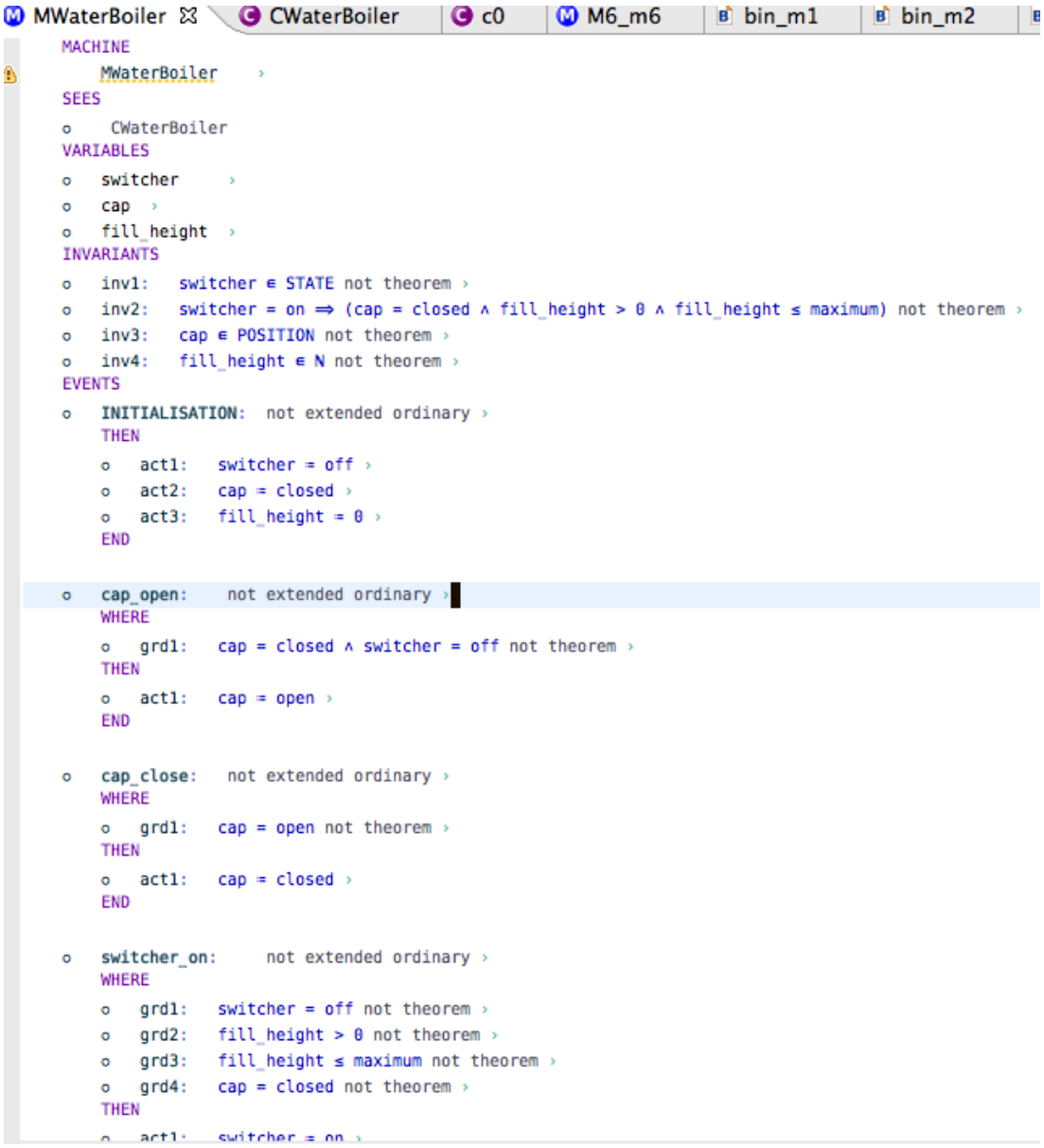


图5.3 热水器machine界面

开头的MWaterBoiler是机器名称，接着是在VARIABLES中定义多个变量，在ENVARIANTS中定义多个不变式,不变式是所有事件必须保持的条件,在INETIALISATION中初始化每个变量，然后就可以在EVENTS中定义编写各个事件了。可以在工具栏中点击

 这些按钮来创建变量，不变式，事件等。整个过程都是运用集合论和逻辑的知识来编写进行验证的。

分析开关打开事件，首先点击  来创建个时间，然后在事件名处右击来选择要添加的guard和action。Guard是事件发生的先决条件，每个事件只有当满足所有Guards时才能执行action操作。

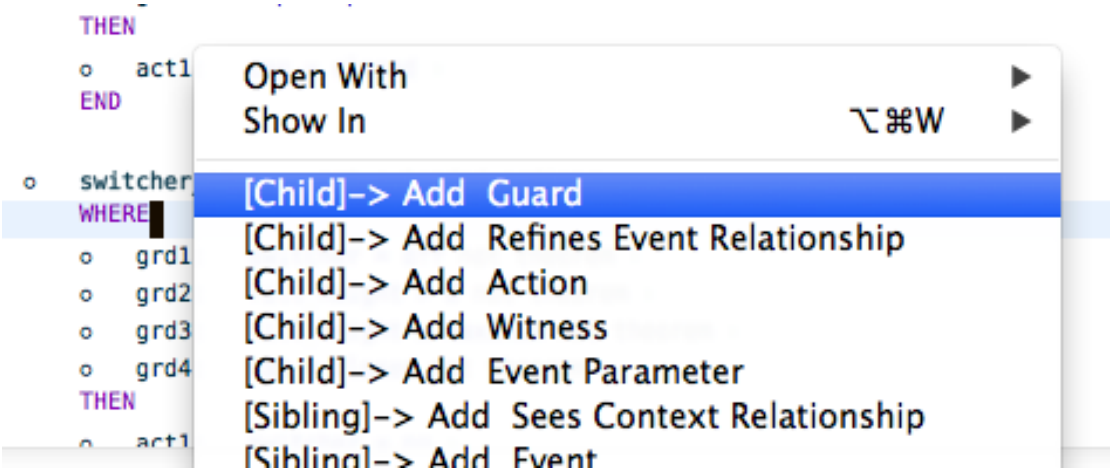
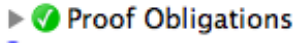


图 5.4 创建事件，卫兵等图示

当建模完成后，要查看有没有错误，如果没有错误，查看证明义务  是否全绿，如果不是代表有些证明没有自动证明需要手动，或者是有自动证明错误的，任然需要手动来修改。只有当证明义务全部都正确时才可以。

Machine 中定义和描述的是项目中动态的部分，可操作可改变的部分，而 Context 中定义和描述的是项目中静态的部分，如一些常量，定量，集合等。下图是热水器的 Context 内容，在其中定义了两个集合，五个常量和六个定量。

```
CONTEXT
    CWaterBoiler     ›
SETS
o   POSITION      ›
o   STATE     ›
CONSTANTS
o   open      ›
o   closed    ›
o   on    ›
o   off   ›
o   maximum   ›
AXIOMS
o   axm1:   POSITION = {open, closed} not theorem ›
o   axm2:   ¬ open = closed not theorem ›
o   axm3:   STATE = {on, off} not theorem ›
o   axm4:   ¬ on = off not theorem ›
o   axm5:   maximum ∈ ℕ1 not theorem ›
o   axm6:   maximum = 3 not theorem ›
END
```

图 5.5　CONTEXT 内容