26-28 сентября 2014, Калуга,
XI конференция разработчиков свободных программ

LinuxTesting.org

Rodin – платформа для разработки и верификации моделей на Event-B

Илья Щепетков
shchepetkov@ispras.ru

ISP RAS

Institute for System Programming of the Russian Academy of Sciences

Критичные по безопасности системы

Ошибки в таких системах могут привести к:

- Гибели и травмам людей
- Крупным финансовым потерям
- Ущербу окружающей среды

Цена ошибки

- Ошибка округления в ПО зенитного ракетного комплекаса Patriot, приводящая к отставанию часов на 300 миллисекунд за 100 часов боевого дежурства. Итог – пропущенная иракская баллистическая ракета Scud, 28 убитых и >100 раненых, 1991 г.
- Переполнение при преобразовании 64bit float -> 16bit int в ПО инерциальной навигационной системы ракеты Ариан 5 – взрыв ракеты спустя 40 секунд после запуска, потеря $500млн, 1996 г.
- Ошибка в программе переноса БД доноров органов в Великобритании — у 25 человек взяли не те органы, 2009 г.
- Ошибки в ПО автомобиля Toyota Prius: компания отозвала более миллиона автомобилей из-за ошибки в алгоритме управления ABS, 2010 г.

Формальные методы

Формальные методы могут быть использованы для достижения максимальной уверенности в отсутствии ошибок в системе.

Использование формальных методов состоит из:

- Разработки математической модели системы и требований к ней
- Доказательства корректности разработанной математической модели

Event-B и Rodin

Event-B – один из наиболее активно развивающихся формальных методов.

Свободная платформа Rodin содержит:

- Текстовый редактор моделей на Event-B
- Набор систем автоматического доказательства
- Поддержку интерактивного доказательства

многочисленные плагины

Пример

Рассмотрим систему управления охлаждением ядерного реактора.

Описание системы: вода для охлаждения реактора либо не подаётся вовсе, либо подаётся из близлежащего водоёма, либо – в черезвычайных ситуациях – из специального резервуара с охлаждённой водой.

Требование: вода из водоёма и охлаждённая вода не должны поступать одновременно.

Пример

Соответствующая модель на Event-B выглядит так:

```
machine CoolingSystem

variables normal_water_flows cold_water_flows

invariants
  @normal_water_flows_type normal_water_flows ∈ BOOL
  @cold_water_flows_type cold_water_flows ∈ BOOL
  @CoolingSystemIsSafe
      ¬(normal_water_flows = TRUE ∧ cold_water_flows =
TRUE)

events
  event INITIALISATION
    then
      @act1 normal_water_flows ≔  FALSE
      @act2 cold_water_flows ≔  FALSE
  end

  event start_normal_water
    where
      @grd1 cold_water_flows = FALSE
    then
      @act1 normal_water_flows ≔ TRUE
  end

  . . .
```

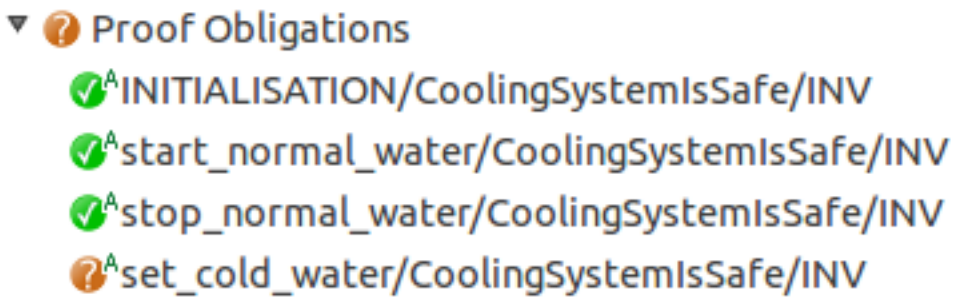
```
  . . .
  event stop_normal_water
    then
      @act1 normal_water_flows ≔ FALSE
  end

  event set_cold_water
    any new_value
    where
      @grd1 new_value ∈ BOOL
    then
      @act1 cold_water_flows ≔
new_value
  end
end
```

Требуется доказать, что инварианты не нарушаются в результате выполнения событий.

Пример

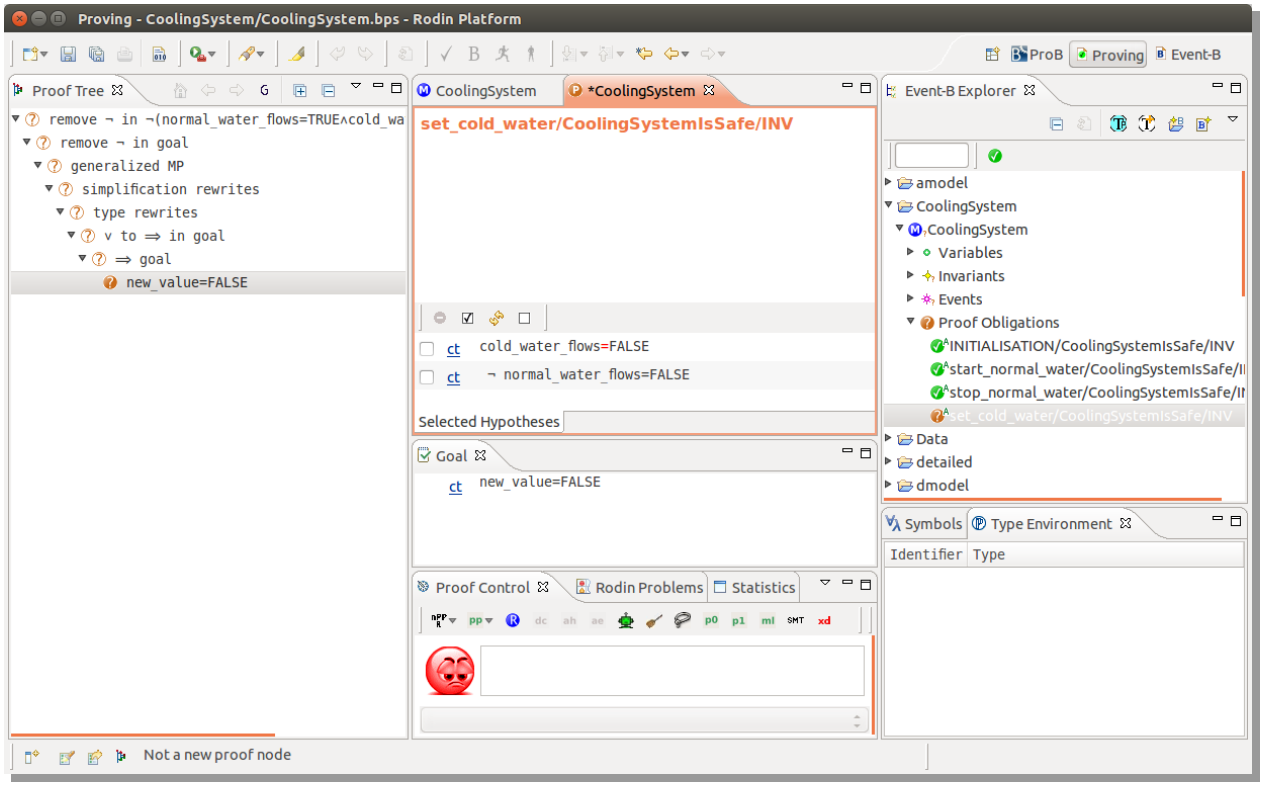
Rodin выявляет все случаи, требующие доказательства, а также для некоторых из них проводит его автоматически.



В данном случае Rodin не смог доказать сохранность инварианта *CoolingSystemIsSafe* для события set_cold_*water*.

Пример

В случае неудачи систем автоматического доказательства можно воспользоваться интерактивным доказательством.



Мы не можем доказать требуемое – в нашей модели ошибка!

Пример

Для исправления модели достаточно добавить дополнительное предусловие в событие *set_cold_water*.

```
machine CoolingSystem

variables normal_water_flows cold_water_flows

invariants
  @normal_water_flows_type normal_water_flows ∈ BOOL
  @cold_water_flows_type cold_water_flows ∈ BOOL
  @CoolingSystemIsSafe
      ¬(normal_water_flows = TRUE ∧ cold_water_flows =
TRUE)

events
  event INITIALISATION
    then
      @act1 normal_water_flows ≔  FALSE
      @act2 cold_water_flows ≔  FALSE
  end

  event start_normal_water
    where
      @grd1 cold_water_flows = FALSE
    then
      @act1 normal_water_flows ≔ TRUE
  end

                    . . .
```

```
                    . . .
  event stop_normal_water
    then
      @act1 normal_water_flows ≔ FALSE
  end

  event set_cold_water
    any new_value
    where
      @grd1 new_value ∈ BOOL
      @grd2 new_value = TRUE
                ⇒ normal_water_flows =
FALSE
    then
      @act1 cold_water_flows ≔ new_value
  end
end
```

Пример индустриального применения

14-я полностью автоматическая линия парижского метро была разработа с применением формального метода B (предшественник Event-B).

- Была написана и полностью доказана модель на B размером свыше 110 000 строк
- На её основе было сгенерировано 86 000 строк кода на языке Ada
- С момента завершения доказательства не было найдено ни одной ошибки – ни после дополнительных работ по верификации и валидации системы, ни с момента запуска линии в эксплуатацию (1998). Программное обеспечение до сих пор версии 1.0.

Наш опыт использования

С помощью Event-B и Rodin мы:

- Формализовали модель управления правами доступа и информационными потоками операционной системы Astra Linux Special Edition.
- Полностью доказали её корректность
- Обнаружили и исправили ряд неточностей в её изначальном текстовом описании

Плюсы и минусы

- Нахождение ошибок, которые иначе не были бы найдены
- Повышение уверенности в корректности системы
- Решение задачи сопровождения системы при последующих правках и расширениях
- Плагины

- Высокий уровень затрат на проведение формальной верификации
- Ограниченная поддержка Rodin командной разработки
- Отсутствие поддержки выделения часто используемых выражений в отдельные сущности с последующим доступом к ним по ссылке

LinuxTesting.org

Вопросы?

Илья Щепетков
shchepetkov@ispras.ru

ISP RAS
Institute for System Programming of the Russian Academy of Sciences