Лекция 4. Язык Event-B

Цель лекции

Изучить язык Event-B и среду моделирования Rodin.

Содержание

Почему Event-B

- Сертификация по 4 уровню доверия к СЗИ требует разработать формальную модель управления доступом и верифицировать ее с применением инструментальных средств
- ГОСТ Р 59453.X-2021 Защита информации. Формальная модель управления доступом
- Event-B/Rodin - основные средства моделирования и верификации формальных моделей управления доступом

Свойства Event-B/Rodin

- формальный
- статически типизированный
- минималистичность
- математическая основа — теория множеств
- наличие инструментальных средств для написания моделей на Event-B и верификации (Rodin), анимации (ProB) и др.

Дискретно-событийное моделирование

- система обладает состоянием
- событие - воздействие на систему, система мгновенно дает отклик и может изменить свое состояние
- события упорядочены
- система может включать программы, аппаратуру, внешнюю среду

Моделирование - это не программирование

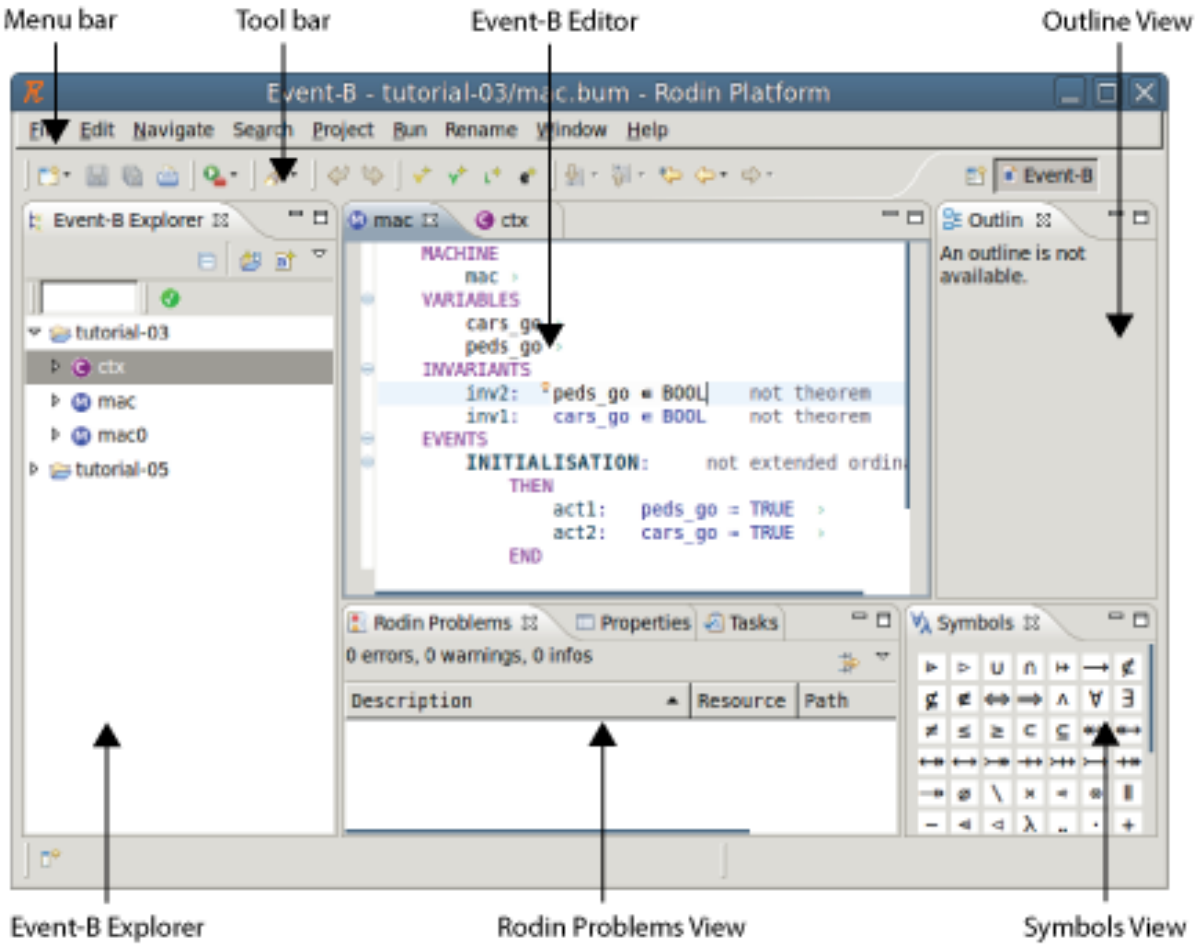
Цель этой деятельности - не написание алгоритма/программы на "еще одном языке а получение правильных требований к программе/системе:

1. Точные, недвусмысленные формулировки
2. Абстракция (отсутствие лишних деталей $\rightarrow$ проще)
3. Доказана непротиворечивость
4. Возможна поэтапная разработка модели (добавление информации небольшими порциями с постоянным доказательством корректности)

Основные элементы формального метода Event-B

- статические свойства системы - ‘context‘
- динамические свойства системы - ‘machine‘
- формальная модель - множество контекстов и последовательность машин
- контест может «расширять» другие контексты
- машина может «видеть» несколько контекстов
- машина может «уточнять» другую машину

Как это выглядит в Rodin



Содержание

Контекст

- название контекста
- расширяемые контексты-предки
- несущие множества
- константы
- аксиомы/теоремы

Машина

- название машины
- уточняемая абстрактная машина
- контексты
- переменные
- инварианты
- события (в т.ч. INITIALISATION)

Событие

- название события
- уточняемое абстрактное событие
- параметры
- охранные условия
- действия

Содержание

Простейшие типы данных

- `INT`
- `NAT`
- `NAT1`
- `BOOL`
- несущие множества

Выражения — общие положения

- статически типизированы
- могут быть неопределены (пример: делят на 0), есть wd-условие
- логика первого порядка: предикаты и выражения синтаксически различаются (значением переменной не может быть предикат, но может быть функция)

Логические выражения

- константы `TRUE`, `FALSE`
- логические связки `&`, `or`, `=>`, `<=>`, `not`
- логические связи ленивые, при использовании различных связок в одном выражении скобки необходимы
- квантор всеобщности `!`
- квантор существования `#`
- сравнения `=`, `/=`, `>`, ...
- принадлежность `:`, `/:`, `<<`, ...

Выражения с множествами

- пустое множество `{}`
- перечисление `{1, 2, 3}`
- генератор `{x * 2 | x : 1 .. 10}`
- целочисленные диапазоны (обе границы включаются)
  `a .. b`
- теоретико-множественные операции \/, /\, \

Предопределенные функции и предикаты

- предикат как выражение типа BOOL `bool(predicate)`
- конечность множества `finite(set)`
- размер множества `card(set)`
- равенство объединению непересекающихся множеств `partition(set, set1, set2, ..., setN)`

Пары и отношения

- пара `a |-> b`
- декартово произведение `A ** B`
- множество подмножеств `POW(A)`, `POW1(A)`
- произвольное отношение `A <-> B`
- тотальное отношение `A <<-> B`
- сюръективное отношение `A <->> B`
- тотальное сюръективное отношение `A <<->> B`

Операции над отношениями

- область определения (домен) `dom(relation)`
- область значений `ran(relation)`
- проекция `relation[set]`
- оставить только поддомен `set <| relation`
- убрать поддомен `set <<| relation`
- оставить только значения `relation |> set`
- убрать значения `relation |>> set`

Функции

Функция — это детерминированное подмножество отношения.

- множество тотальных функций `A -> B`
- множество частичных функций `A +-> B`
- применение `function(arg)`

Действия

- одиночное `x := E`
- множественное `x1, x2, ..., xN := E1, E2, ..., EN`
- замена пары в отношении `x(F) := E`
- неявное одиночное
  `x :| before-after-predicate with x and x'`
- неявное множественное
  `x1, ..., xN :| b-a-p with x1, ..., xN, x1', ..., xN'`
- неявный выбор из непустого множества `x :: set`

Содержание

Функционирование машины

1. состояние машины — значения всех переменных
2. событие — переход из одного состояния в другое
3. событие может иметь параметры
4. переход возможен, только если истинны все охранные условия
5. в этом случае выполняются все действия
6. действия одного события выполняются одновременно
7. начальное состояние — результат события `INITIALISATION`

Корректно определенная машина/контекст

1. `WD`: нет неопределенных выражений
2. `INV`: не нарушены инварианты
3. `THM`: не нарушены теоремы
4. `FIS`: неявные действия разрешимы
5. `GRD`, `SIM`, `EQL`: уточнение корректно

Rodin генерирует условия верификации для формального доказательства, что машина корректно определена.