Méthodes formelles

Mohamed Tounsi

Faculté des Sciences Économiques et de Gestion de Sfax

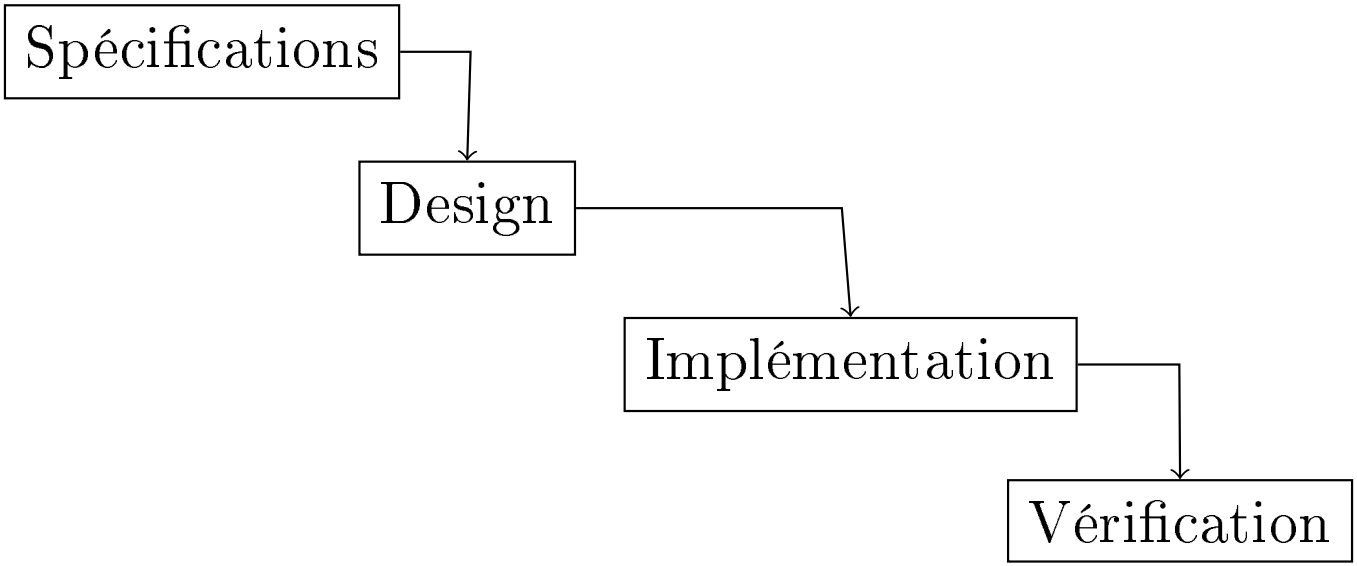
Avril 2015

Resources

- Rodin pour les débutants:
  http://handbook.event-b.org/current/html/tutorial.html
- Résumé des notions Event-B:
  http://wiki.event-b.org/images/EventB-Summary.pdf
- Livre de *Jean-Raymond Abrial*:
  Modeling in Event-B: System and Software Engineering
- Examples d'Event-B:
  www.stups.uni-duesseldorf.de/bmotionstudio/index.php/User_Guide/Tutorial

Introduction

Construire un logiciel: le modèle en cascade



Question

Ce modèle est il adapté aux systèmes “complexes” ou “critiques” ?

Introduction

Quelques exemples d'échecs

Therac 25: machine de radiothérapie (1985)

- comportement incorrect du matériel, provoquant l'irradiation sévère de plusieurs patients.
- système de contrôle écrit en assembleur.
- au moins trois morts.

Arianne 5 (1996)

- conversion flottant 64 bits en entier 16 bits cause une exception processeur.
- réutilisation des programmes de contrôle d'Ariane 4.
- code inutile dans le cas d'Arianne 5.
- environ 500 millions d'euros.

Introduction

Une approche imparfaite

Constatations

- La validation de chaque étape est manuelle ou “semi-formelle”
- Les spécifications sont-elles correctes ?
- Les tests correspondent-ils aux spécifications ?
- Les tests sont-ils exhaustifs ?

Introduction

La vérification formelle

Principe

- Un programme correspond à un objet mathématique. Cet objet sera utilisé pour prouver la *correction* du programme.
- L'approche de "theorem proving" consiste à énoncer des propositions et à les démontrer dans un système de déduction du calcul des prédicats.
- La vérification des programmes consiste à démontrer les théorèmes de correction et de complétude. Ces théorèmes permettent de formaliser le comportement attendu d'un programme et leurs démonstration permet d'en affirmer leur vérité. (exemple: calcul de la plus faible précondition)

La méthode Event-B

Une méthode de développement formelle *intégrée*

- spécifications abstraites,
- Le *raffinement* guide la preuve et le développement,
- génération de code automatique.

Ayant fait ses preuves dans les milieux industriels

- Ferroviaire: Ligne Météor, Val de Roissy CDG, etc.
- Applications “cartes à puces”: GemPlus, Schlumberger

Le développement de modèles avec Event-B

- Event-B n'est pas un langage de programmation,
- Event-B est une notation utilisée pour développer des modèles mathématiques des systèmes de transitions discrètes,
  - programme séquentiel,
  - programme distribué,
  - programme concurrent,
  - circuits électroniques
  - ....
- Event-B doit être utilisé conjointement avec la plate-forme Rodin

Historique

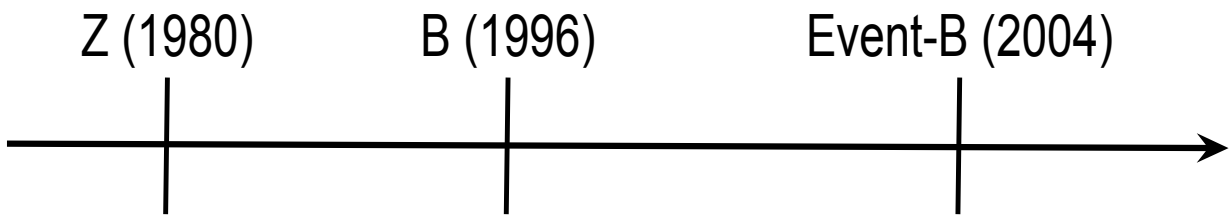


Figure : Évolution des langages Z, B et Event-B

langage Z (Z-EVES)

Un formalisme pour la spécification, la conception et l'écriture des programmes, visant a permettre a l'utilisateur de conserver a tous les niveaux de l'analyse la maîtrise complète des processus en jeu *[Les langages de spécification 1979]*

Différence entre B et Event-B

Méthode B (atelier B)

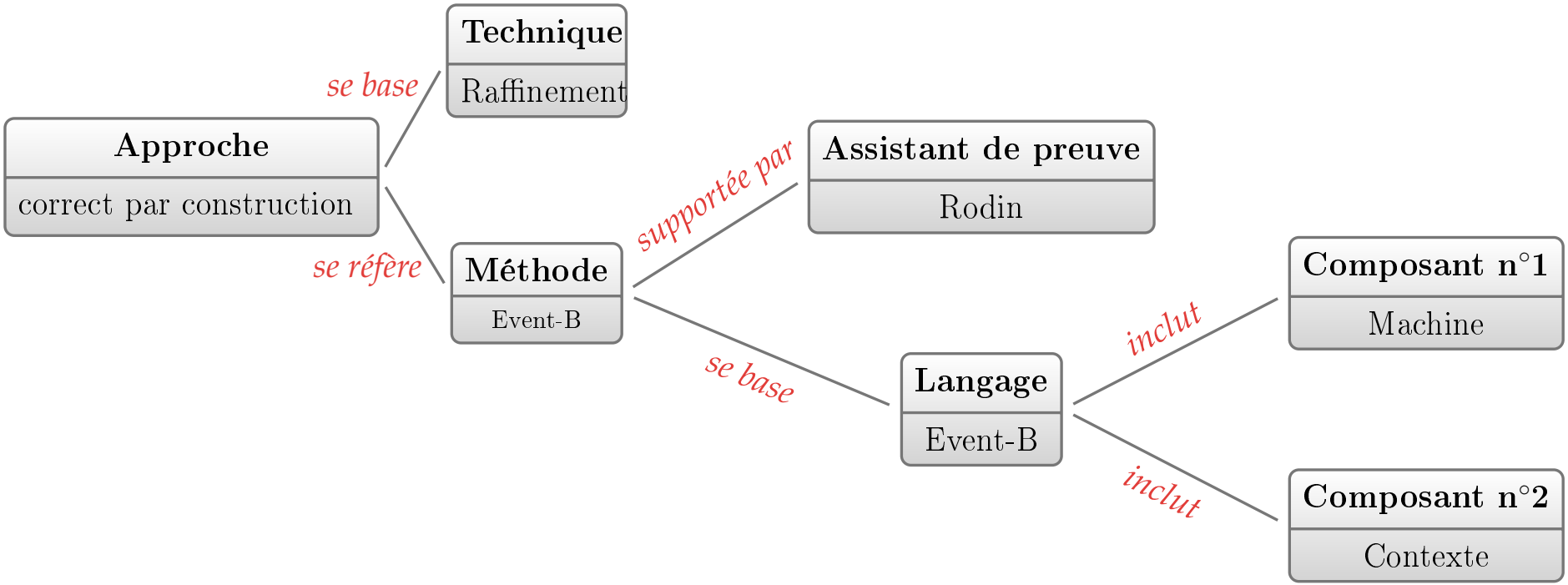
La méthode B est adaptée au développent des programmes impératives utilisant des structures de contrôle classiques: boucles, séquences, conditionnelles, etc.

Méthode Event-B (Rodin)

La méthode Event-B permet de décrire les systèmes réactifs:

- la structure de contrôle de base est l'événement,
- l'événement peut se produire si sa garde est vraie,
- un système peut ne pas terminer, mais son état doit être toujours correct.

L'approche "correct par construction"

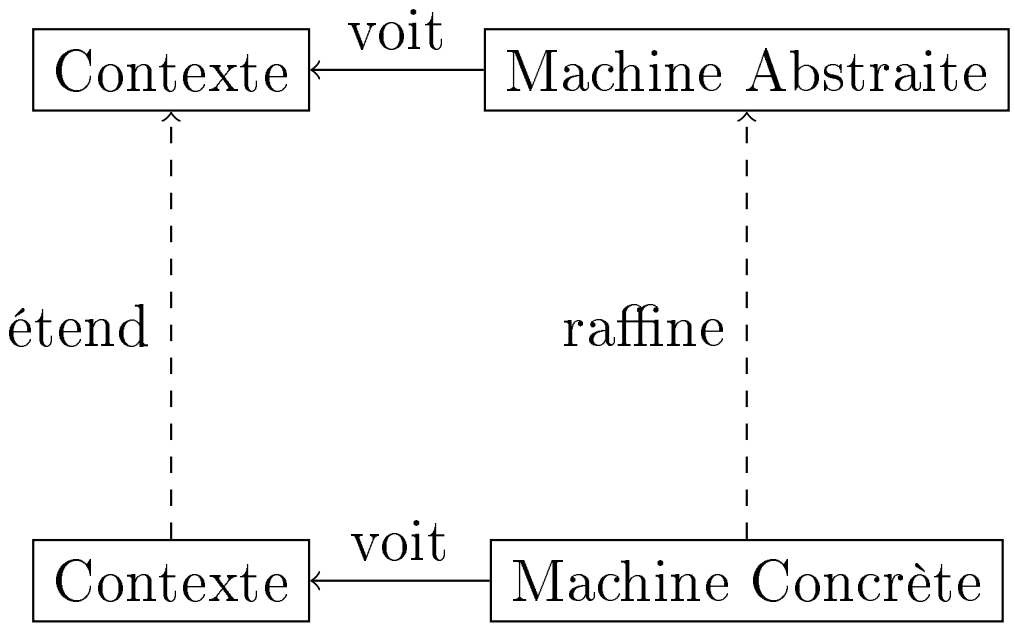


La méthode *Event-B*

- conçue par *Jean-Raymond Abrial* début années 2000,
- évolution de la méthode B classique et du langage Z,
- fondée sur la logique du premier ordre, la théorie des ensembles et le langage de substitutions généralisées.

La méthode Event-B

Langage de Modélisation



Données :	ensembles
Opérations :	substitutions généralisées
Propriétés :	prédicats du premier ordre

La méthode Event-B

Contextes

Un *contexte* représente la partie *statique* d'un programme, par exemple ses constantes, ses paramètres, etc.

CONTEXT

CONTEXT
Le nom du contexte
EXTENDS
Les contextes vus par le contexte
SETS
Les ensembles du contexte
CONSTANTS
Les constantes du contexte
AXIOMS
Les propriétés du contexte
THEOREMS
Les théorèmes du contexte

Figure : Structure du contexte

La méthode Event-B

Contextes: un premier exemple

CONTEXT	Salle	
SETS	ETATS	
CONSTANTES	Ouvert	
	Ferme	
	temp_max	
	temp_min	
AXIOMS	axm1:	ETATS ={Ouvert, Ferme}
	axm2:	temp_ max $\in \mathbb{N}$
	axm3:	temp_ min $\in \mathbb{N}$
	axm4:	temp_min $<$ temp_max

La méthode Event-B

Machines et programmation événementielle

Programmation événementielle

- un programme réactif est un programme répondant à des changements de son *environnement*.
- exemples: systèmes d'exploitation, contrôleurs industriel, etc.

Exemple

```
Répéter indéfiniment
  si température > temp_max-((temp_max-temp_min)/2)
    ouvrir les fenêtres
```

La méthode Event-B

Machines: un premier exemple (incomplet)

MACHINE	Exemple	
SEES	Salle	
VARIABLES	temp fenetre	
INVARIANTS	inv1:	fenetre $\in$ ETATS
	inv2:	temp $\in$ temp_min .. temp_max
EVENTS	Event	Ouvre $\widehat{=}$
		where grd1: fenetre = Ferme
		grd2: temp>temp_max-((temp_max - temp_min)/2)
		then act1: fenetre := Ouvert

La méthode Event-B

Anatomie d'une machine

La machine représente la partie *dynamique* d'un programme,

MACHINE
Le nom de la machine
REFINES
Le nom de la machine à raffiner
SEES
Les contextes vus par la machine
VARIABLES
Les variables de la machine
INVARIANTS
Les propriétés de la machine
VARIANT
Une mesure de preuve
EVENTS
Les événements de la machine

Figure : Structure d'une machine

La méthode Event-B

Anatomie d'un événement

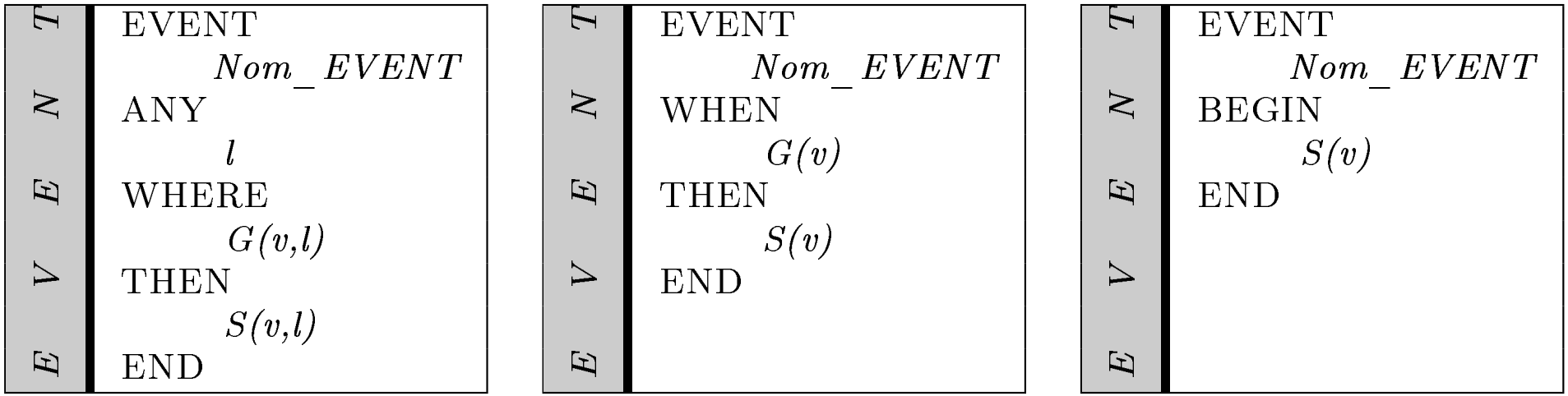


Figure : Différents types d'événements

- des variables l: dont la porté est restreinte à l'événement,
- des gardes G: les prédicats spécifient l'état dans lequel l'événement est déclenchable. v sont des variables de la machine.
- des actions S.

La méthode Event-B

Les actions d'Event-B

- Une action décrit les façons avec laquelle une ou plusieurs variables d'état sont modifiées par l'activation d'un événement,
- Une action peut être déterministe ou non-déterministe

affectations déterministes: $x_0, \ldots, x_n := e_0, \ldots, e_n$

exemple : $s := s \cup \{x \mapsto y\}$

affectations non déterministes_1: $e :\in E$

exemple: $temp :\in temp\_min \,..\, temp\_max$

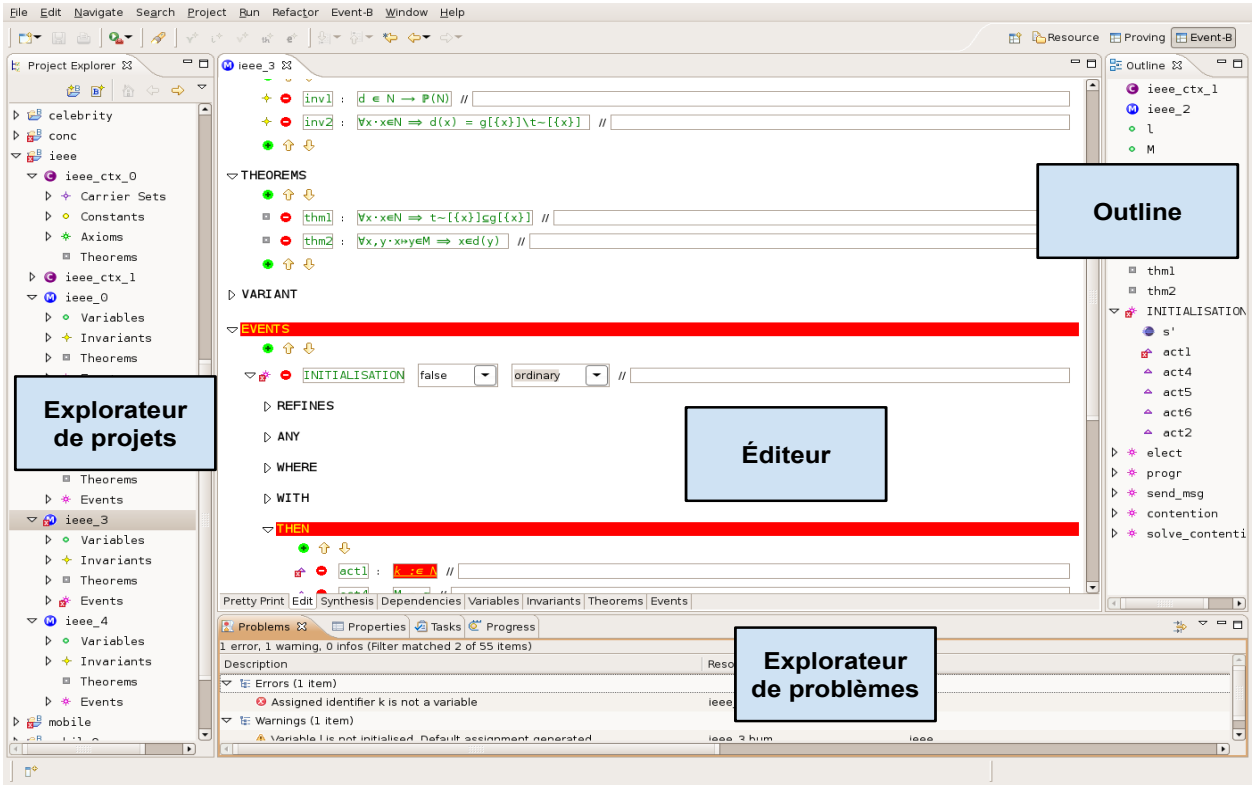
affectations non déterministes_2: $e : |P(e')$

exemple 1: $s : |s' = s \cup \{x \mapsto y\}$
exemple 2: $temp : |temp' \in temp\_min \,..\, temp\_max$

Rodin

Event-B perspective

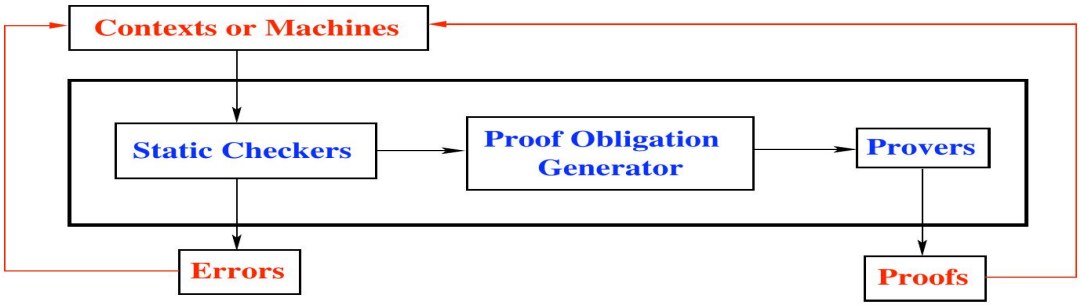
- Rodin est une plate-forme de spécification et de preuve en Event-B,
- Rodin est basée sur Eclipse et peut être étendue à l'aide de plug-ins.



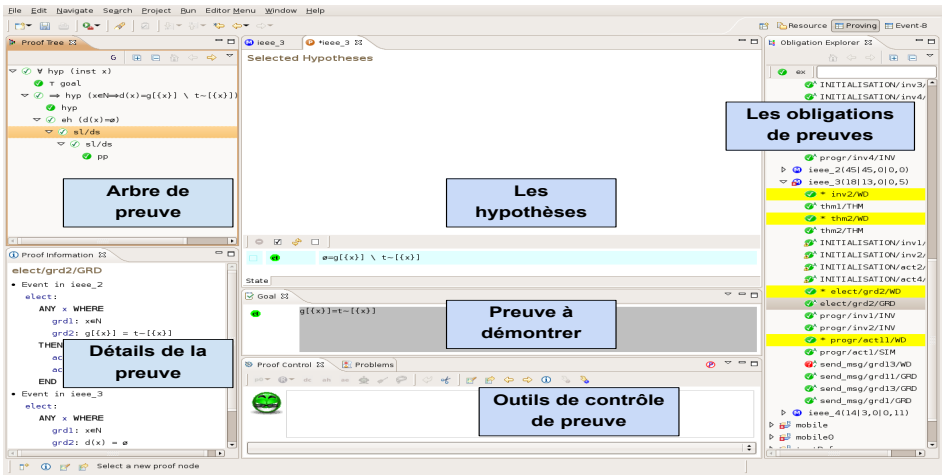
Rodin

Proving perspective

L'étape de la preuve influe beaucoup sur la spécification.



L'agencement des fenêtres n'est pas fixe !



Rodin

Un premier exemple: La bouilloire électrique

On désire spécifier, avec Event-B, le système de la bouilloire électrique. Cette dernière fonctionne de la façon suivante:

1. Ouvrir le couvercle et remplir la bouilloire avec de l'eau froide. A savoir on ne peut pas remplir la bouilloire plus que sa capacité maximale,
2. Démarrer la bouilloire pour faire bouillir l'eau,
3. Arrêter l'appareil et ouvrir le couvercle et verser de l'eau chaude. Toutefois, on peut ne pas verser de l'eau chaude et ajouter de l'eau froide si on n'a pas encore atteint la capacité maximale de la bouilloire.