



IMT Atlantique

Bretagne-Pays de la Loire
École Mines-Télécom

MÉTAMODÉLISATION DU LANGAGE B INTÉGRATION D'EVENT-B À OPENFLEXO

PROJET S5 F2B

Armand Foucalt

Encadrant : Jean-Christophe Bach

SOMMAIRE

1. CONTEXTE DU PROJET

- 1.1 Le projet Formose
- 1.2 La méthode Event-B
- 1.3 L'assistant logiciel Rodin
- 1.4 OpenFlexo
- 1.5 Objectifs du projet

2. L'API DE RODIN

- 2.1 Architecture générale
- 2.2 Gestion de la méthode B
- 2.3 Gestion des projets Rodin
- 2.4 Synthèse : les fonctionnalités utiles

3. CONCEPTION DU PLUGIN

- 3.1 Fonctionnalité de l'abstraction
- 3.2 Implémentation de l'abstraction
- 3.3 Écoute des instructions
- 3.4 Exécution des instructions



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

PARTIE 1

CONTEXTE DU PROJET



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

OpenFlexo comme passerelle entre les deux univers



Spécification formelle de systèmes

- Description d'un système
 - Déclaration de propriétés et d'évènements
 - Déclaration de contraintes
- Validation formelle du système : les contraintes sont-elles respectées ?
- Raffinement du modèle : abstraction → implémentation

Un assistant à la conception et à la preuve

- Développé avec le framework Eclipse
 - Conçu comme un jeu de plugins pour Eclipse
- Open-source : l'API est libre et accessible
 - Fonctionnalités de gestion de projet, de manipulation d'éléments
 - Calcul logique de preuves

Event-B Explorer

- tutorial-03
 - Context1
 - Machine1
 - mac**
 - Variables
 - Invariants
 - Events
 - Proof Obligations
 - INITIALISATION/inv3/INV
 - set_peds_go/grd1/THM
 - set_peds_go/inv3/INV
 - set_peds_stop/inv3/INV
 - set_cars/inv3/INV

```

1 MACHINE
2   mac
3 VARIABLES
4   cars_go
5   peds_go
6 INVARIANTS
7   inv1: cars_go ∈ BOOL not theorem
8   inv2: peds_go ∈ BOOL theorem
9   inv3: ~ (cars_go = TRUE ∧ peds_go = TRUE) not theorem
10  inv4: ⊤ not theorem
11 EVENTS
12  INITIALISATION: not extended ordinary
13  THEN
14    act1: cars_go = FALSE
15    act2: peds_go = FALSE
16  END
17
18  set_peds_go: not extended ordinary
19  WHERE
20    grd1: cars_go = FALSE theorem
21  THEN
22    act1: peds_go = TRUE
23  END
24
25  set_peds_stop: extended ordinary
26  THEN
27    act1: peds_go = FALSE
28  END
29
30  set_cars: not extended ordinary
31  ANY
32    new_value
33  WHERE
34    grd1: new_value ∈ BOOL not theorem
35    grd2: new_value = TRUE ⇒ peds_go = FALSE not theorem
36  THEN
37    act1: cars_go = new_value
38  END
39
40 END
41

```

Outline

An outline is not available.

Rodin Problems Properties Tasks

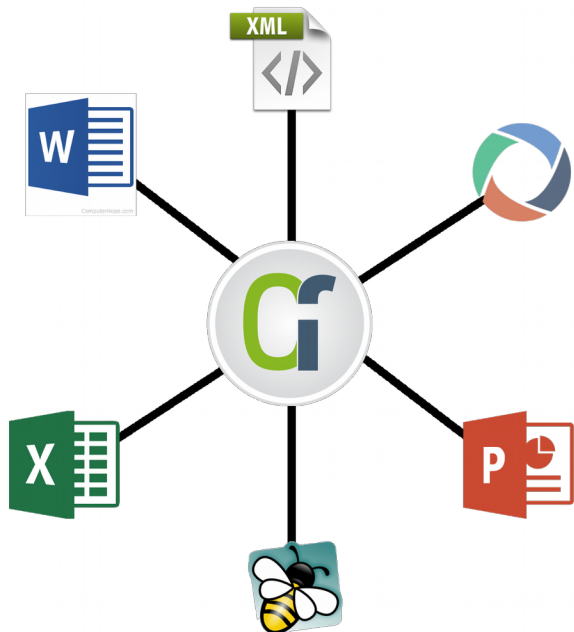
0 items

Description	Resource	Path	Location	Type

Symbols

Navigation and editing symbols

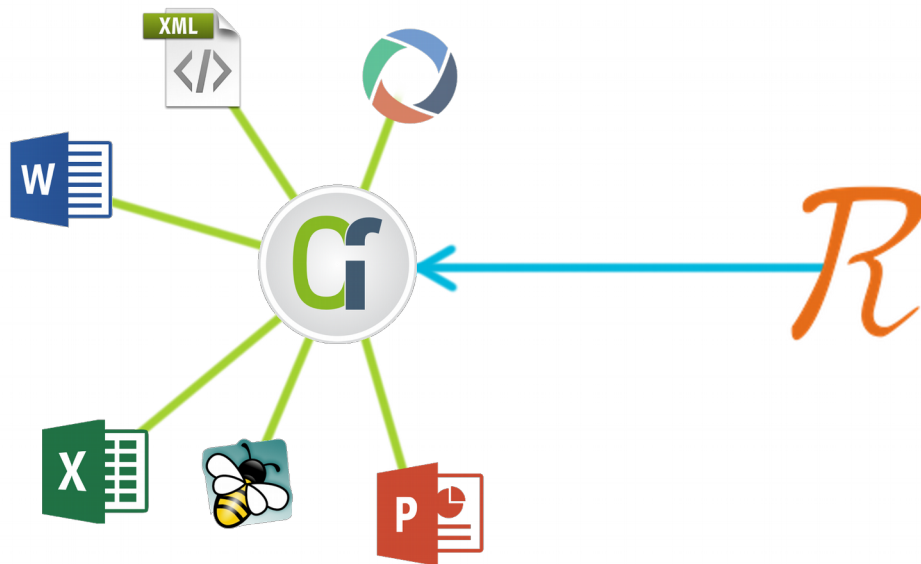
OpenFlexo comme pont



- Infrastructure logicielle
 - Free Modeling Editor
 - Viewpoint Modeler
 - Smartdocs
 - View Editor
- Fédération de modèles
 - Multiples modèles
 - Multiples vues / contrôleurs

1.5 Objectif du projet

Intégration de la méthode B à OpenFlexo



Intégration de la méthode B à OpenFlexo

- Étude et analyse de l'API de Rodin
 - Quelles sont les fonctionnalités intéressantes ?
- Conception d'une abstraction de l'API de Rodin
 - Synthèse des fonctionnalités intéressantes
- Utilisation de l'abstraction depuis OpenFlexo

PARTIE 2

L'API DE RODIN



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2.1 Architecture générale

Deux packages principaux

- `org.eventb.XXX`
 - Gestion de la méthode B
 - Création d'évènements, d'actions, de variables...
- `org.rodinp.XXX`
 - Gestion des projets Rodin en tant que type de projets Eclipse
 - Création de projets et de machines (`.bum`)

Analyse des fonctionnalités constructrices

- Tous les types d'éléments sont définis par un protocole
 - Interfaces définies dans `org.eventb.core`
 - `IEvent` → événements
 - `IAction` → actions
 - `IGuard` → gardes
 - ...
 - Implémentation des interfaces dans `org.eventb.core.basis`
 - Pas prévue pour être utilisée → Manipulation à travers les interfaces

L'exemple des évènements

- Définis par l'interface `IEvent`, implémentés par la classe `Event`
 - Accesseurs / constructeurs pour les types enfants
 - `getAction`, `getGuard`, `getParameter...`
 - Méthodes constructrices : renvoient toujours une instance
 - Il faut écrire ces enfants dans le projet
 - Création par appel de `getEvent` sur le parent

Les projets et la base de données Rodin

- Base de données Rodin `RodinDB`
 - Conserve la hiérarchie d'éléments du projet
- Projets Rodin : l'interface `IRodinProject`
 - Fait référence à un `IProject` d'Eclipse
 - Création à partir de la `RodinDB`
- Fichiers Rodin : l'interface `IRodinFile`
 - Création de machines (`.bum`), de contextes (`.buc`)

Fonctionnalités à extraire : fonctionnalités constructrices essentielles

- Création de projet Rodin
- Création de machine : fichier Rodin .bum
- Ajouts d'éléments de méthode B
 - Évènement dans une machine
 - Action, garde dans un évènement

PARTIE 3

CONCEPTION DU PLUGIN



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

3.1 Fonctionnalités de l'abstraction

Six fonctions principales

Objectif : proposer les fonctionnalités constructrices de l'API

- IrocinProject `createRodinProject(projectName)`
- ImachineRoot `createMachine(projectName, machineName)`
- Ievent `addEvent(machineName, eventName)`
- Iaction `addAction(eventName, actionName)`
- Iguard `addGuard(eventName, guardName)`
- Ivariable `addVariable(machineName, variableName)`

Développement en mode plugin

Accès au workspace Eclipse (projets) et aux autres plugins

Fonctionnement de l'abstraction : la classe `ApiAbtractor`

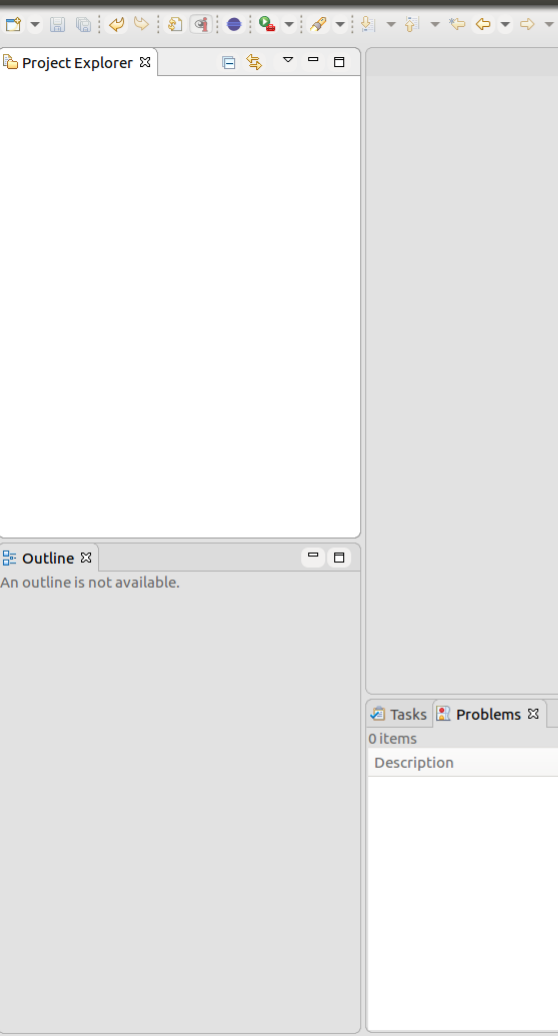
- Interface publique de l'abstraction
 - Fournit les méthodes de création d'éléments
 - Prennent des noms en arguments pour faciliter la communication
 - Initie la connexion avec le workspace et la RodinDB
- Fonctionnement interne
 - Maintient à jour l'environnement des éléments créés
 - Exécution d'une instruction : recherche dans l'environnement par nom

Écoute d'une socket : la classe `InstructionListener`

- Implémentation d'un serveur TCP
 - Attente d'une connexion, puis écoute d'instructions
 - Modèle client-serveur, avec un client unique
 - Écoute sur un thread secondaire : `extends Thread`
 - Messages : `operation;arg1;arg2 ;...\n`
 - Transmission des messages à `InstructionDispatcher`

Exécution des instructions : la classe `InstructionDispatcher`

- Définition des opérations dans l'énumération `Operation`
 - `CREATE_PROJECT`, `ADD_EVENT...`
- Point d'entrée : `execute`
 - Parsage d'une instruction sous la forme d'un `String`
 - Passage de la version parsée à `dispatchInstruction`
 - Appel de la fonction correspondante d'`ApiAbstractor`



Project Explorer

Outline X

An outline is not available.

0 items selected

Rodin - Connected to localhost:20001

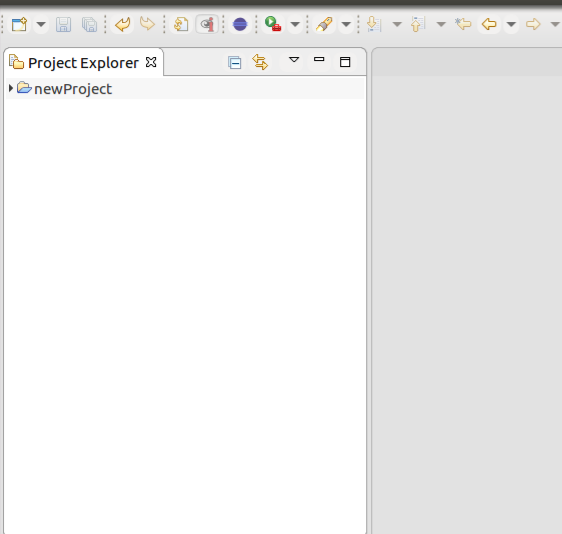
```
Now connected to localhost:20001
```

Type in your message

➤

Tasks Problems

0 items



An outline is not available.

```
Rodin - Connected to localhost:20001
Now connected to localhost:20001
Type in your message
> 1;newProject
Type in your message
> 
```

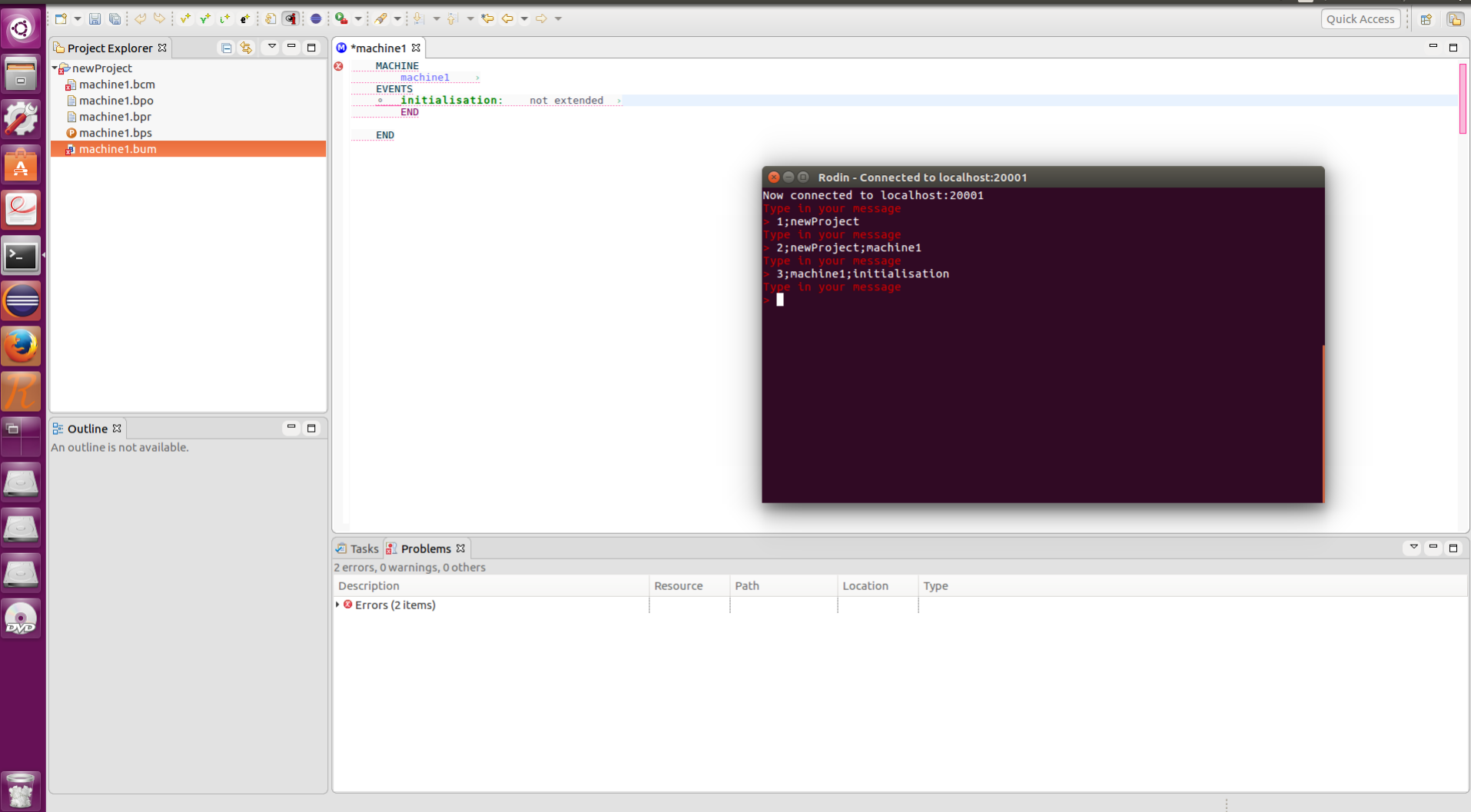
The screenshot shows an IDE interface with the following components:

- Project Explorer:** Displays a project named 'newProject' with files 'machine1.bcm', 'machine1.bpo', 'machine1.bpr', 'machine1.bps', and 'machine1.bum'. The file 'machine1.bum' is selected.
- Code Editor:** Shows the content of 'machine1.bum', which is a machine code snippet:

```
MACHINE
machine1
END
```
- Terminal:** A window titled 'Rodin - Connected to localhost:20001' is open, showing the following output:

```
Now connected to localhost:20001
Type in your message
> 1;newProject
Type in your message
> 2;newProject;machine1
Type in your message
>
```
- Outline:** A panel titled 'Outline' is visible, stating 'An outline is not available.'
- Problems:** A panel titled 'Problems' shows '2 errors, 0 warnings, 0 others'. It contains a table with the following structure:

Description	Resource	Path	Location	Type
2 Errors (2 items)				



Project Explorer

- newProject
 - machine1.bcm
 - machine1.bpo
 - machine1.bpr
 - machine1.bps
 - machine1.bum

Outline

An outline is not available.

Tasks Problems

2 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
✖ Errors (2 items)				

Rodin - Connected to localhost:20001

```
Now connected to localhost:20001
Type in your message
> 1;newProject
Type in your message
> 2;newProject;machine1
Type in your message
> 3;machine1;initialisation
Type in your message
>
```

Project Explorer

newProject

machine1.bcm

machine1.bpo

machine1.bpr

machine1.bps

machine1.bum

Outline

An outline is not available.

*machine1

```
MACHINE
  machine1
  EVENTS
    initialisation: not extended
  THEN
    do_something:
  END
END
```

Rodin - Connected to localhost:20001

Now connected to localhost:20001
Type in your message
> 1;newProject
Type in your message
> 2;newProject;machine1
Type in your message
> 3;machine1;initialisation
Type in your message
> 4;initialisation;do_something
Type in your message
>

Tasks Problems

2 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
✖ Errors (2 items)				

The screenshot displays the Rodin IDE interface. On the left, the **Project Explorer** shows a project named **newProject** with files **machine1.bcm**, **machine1.bpo**, **machine1.bpr**, **machine1.bps**, and **machine1.bum**. Below it, the **Outline** panel indicates that an outline is not available.

The main editor area shows a BPN diagram for ***machine1**. The diagram includes a **MACHINE** block containing an **EVENTS** block. The **EVENTS** block has a **WHERE** clause with **initialisation: not extended** and a **THEN** clause with **do_something**. The **END** label is at the bottom of the diagram.

A terminal window titled **Rodin - Connected to localhost:20001** is open, showing the following commands and responses:

```
Now connected to localhost:20001
Type in your message
> 1;newProject
Type in your message
> 2;newProject;machine1
Type in your message
> 3;machine1;initialisation
Type in your message
> 4;initialisation;do_something
Type in your message
> 5;initialisation;provided
Type in your message
>
```

The **Tasks** and **Problems** panel at the bottom shows 2 errors, 0 warnings, and 0 others. The **Problems** tab is active, displaying a table with columns **Description**, **Resource**, **Path**, **Location**, and **Type**. The table contains one entry: **Errors (2 items)**.

Description	Resource	Path	Location	Type
Errors (2 items)				

The screenshot displays the Rodin IDE interface. On the left, the Project Explorer shows a project named 'newProject' with files 'machine1.bcm', 'machine1.bpo', 'machine1.bpr', 'machine1.bps', and 'machine1.bum'. The Outline pane below it states 'An outline is not available.' The main editor window displays the content of 'machine1.bum', which is a BPN file. The code defines a machine with variables, events, and a provided condition. A terminal window titled 'Rodin - Connected to localhost:20001' is open, showing a series of commands and their outputs. At the bottom, the Tasks and Problems pane shows 2 errors, 0 warnings, and 0 others.

Project Explorer

- newProject
 - machine1.bcm
 - machine1.bpo
 - machine1.bpr
 - machine1.bps
 - machine1.bum

Outline

An outline is not available.

***machine1**

```
MACHINE
  machine1
  VARIABLES
    my_variable
  EVENTS
    initialisation: not extended
    WHERE
      provided: not theorem
    THEN
      do_something
    END
  END
```

Rodin - Connected to localhost:20001

```
Now connected to localhost:20001
Type in your message
> 1;newProject
Type in your message
> 2;newProject;machine1
Type in your message
> 3;machine1;initialisation
Type in your message
> 4;initialisation;do_something
Type in your message
> 5;initialisation;provided
Type in your message
> 6;machine1;my_variable
Type in your message
>
```

Tasks Problems

2 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
✖ Errors (2 items)				

Validation de la POC

- Abstraction de l'API de Rodin
- Plugin écoutant des instructions
- Client léger communiquant avec Rodin
- Jeu du scénario avec succès

La communication entre Rodin et une autre application est validée

Prochaine étape

- Implémenter le côté OpenFlexo de la communication
 - Réaliser un Technology Adapter
 - Présenter dans OpenFlexo les fonctionnalités de l'API

Points à améliorer

- Remplacer l'environnement artificiel par la `RodinDB`
- Enrichir les fonctionnalités de l'API

REMERCIEMENTS

MERCI

QUESTIONS

