

Dedukti Editor : une interface pour Dedukti

Ismaël Lachheb

Encadré par Frédéric Blanqui, Rodolphe Lepigre et Emilio Jesús Gallego Arias

Sup Galilée

ismaellachheb@gmail.com

24 septembre 2018

Sommaire

1 Introduction

- Vérification logicielle
- Dedukti
- Assistant de preuve
- Objectifs

2 Gestion du Projet

3 Choix Technologiques

- État de l'existant
- Language server protocol
- Atom

4 Extension

- Fonctionnalités
- Démonstration
- Implémentation
- Diagramme de cas d'utilisations
- Maintenance

5 Conclusion

Vérification logicielle

Problème

La vérification du bon fonctionnement et de la sécurité de systèmes informatiques.

Exemple

- Centrale nucléaire, Transports
- Sécurité Bancaire

Solutions

- vérifier la correction d'un programme informatique est la même chose que vérifier une preuve
- vérifier le fonctionnement d'un programme avec un vérificateur de preuve

Dedukti

Dedukti est un vérificateur de preuve développé par Deducteam au LSV.

- De petite taille (3000 lignes de code)
- Un système de réécriture de fonctions
- Un système de traduction vers et depuis d'autres systèmes

Pas d'interface

Un programme en ligne de commande.

Preuve

```
Theorem forward_small : (forall A B : Prop, A -> (A->B) -> B).
Proof.
  intros A.
  intros B.
  intros proof_of_A.
  intros A_implies_B.
  pose (proof_of_B := A_implies_B proof_of_A).
  exact proof_of_B.
Qed.
```

Figure – Exemple de Preuve interactive (Coq)

Donner une interface *intuitive* aux utilisateurs de Dedukti.

- Afficher les parties de la preuve vérifiées
- Indiquer les parties de la preuve erronées
- Informer l'utilisateur de la cause de l'erreur
- Informer l'utilisateur de l'état de la preuve
- Rendre la preuve lisible
- Démarage automatique pour les fichiers *.dk*

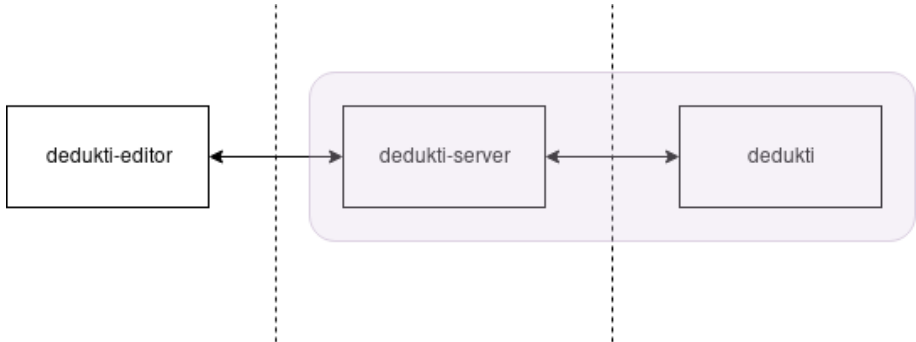
1 Introduction

2 Gestion du Projet

Gestion de Projet

Méthode inspiré de la philosophie Agile

Des réunions régulières pour suivre le projet, fixer des objectifs, proposer des modifications ou des fonctionnalités supplémentaires.



Ismaël Lachheb

Emilio Jesús Gallego Arias

Rodolphe Lepigre

Licence

Open source

Disponible sur :



GitHub

<https://github.com/lachhebo/dedukti-editor>



Atom Store

<https://atom.io/packages/dedukti-editor>

1 Introduction

2 Gestion du Projet

3 Choix Technologiques

- État de l'existant
- Language server protocol
- Atom

État de l'existant

Interface	Assistant de preuve	Éditeur	Synchronisation
Proof-General	Générique (Coq)	Emacs	synchrone
Isabelle Jedit	Isabelle	Jedit	asynchrone
Vscoq	Coq	Vscode	asynchrone

Table – Interfaces existantes

Language Server Protocol

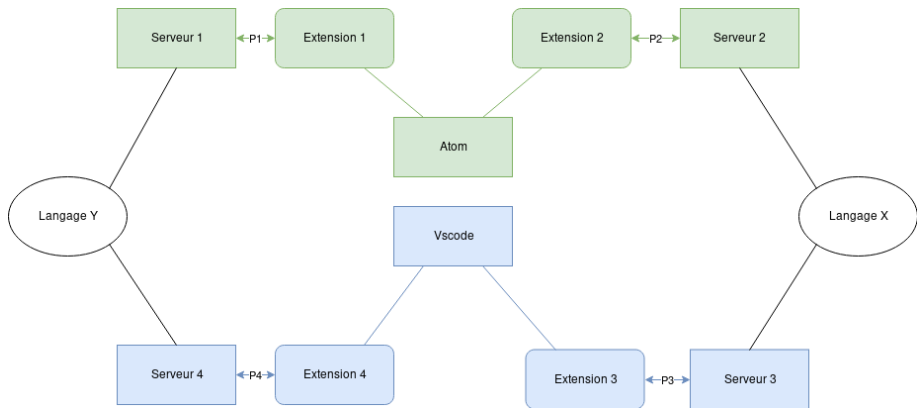
Principe

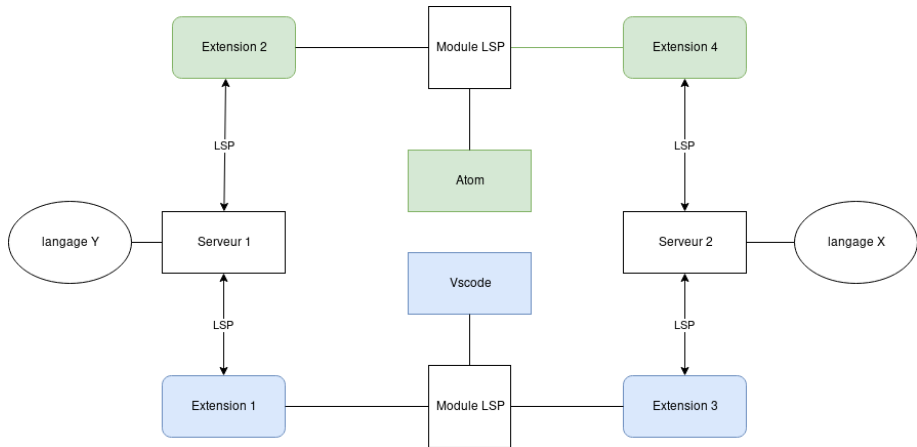
Le *language server protocol* a pour objectif de limiter la redondance de travail à effectuer pour lier par une extension un éditeur de texte à un langage informatique .

- Créé par Microsoft en collaboration avec Red Hat
- Open source
- Première version en Juin 2016

Message	Émetteur	Réceptionneur
publishDiagnostics	serveur	client
didOpen	client	serveur
didChange	client	serveur
completion	serveur	client
hover	serveur	client
codeLens	serveur	client

Table – Exemples de messages issues du protocole.





Avantage	Inconvénients
Bonne documentation Interface intuitive Un marché d'extensions intégré au programme	implémentation de LSP

Table – Atom

1 Introduction

2 Gestion du Projet

3 Choix Technologiques

4 Extension

- Fonctionnalités
- Démonstration
- Implémentation
- Diagramme de cas d'utilisations
- Maintenance

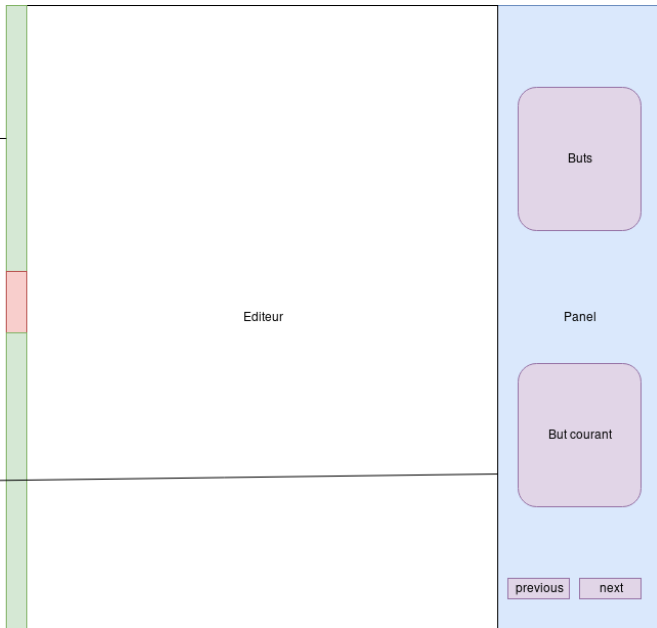
Fonctionnalités

- vérification de la correction d'une preuve
- affichage les messages d'erreurs et leur provenance
- affichage de la liste des buts non résolus selon la position du curseur
- affichage des hypothèses liées au but courant
- boutons et raccourcis claviers pour naviguer dans une preuve
- syntaxe pour les fichiers Dedukti
- gestion des caractères unicode
- système de style

Démonstration

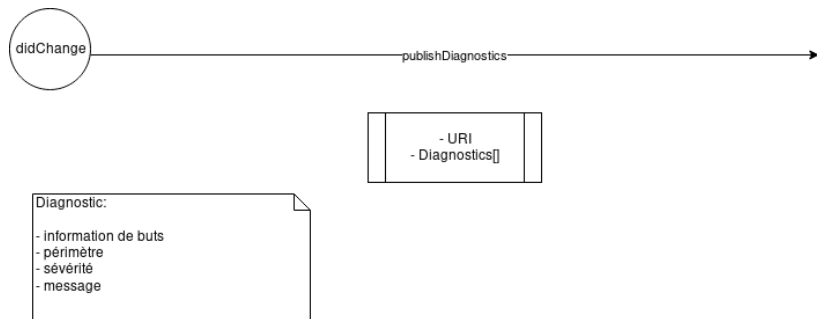
Diagnostics de la preuve

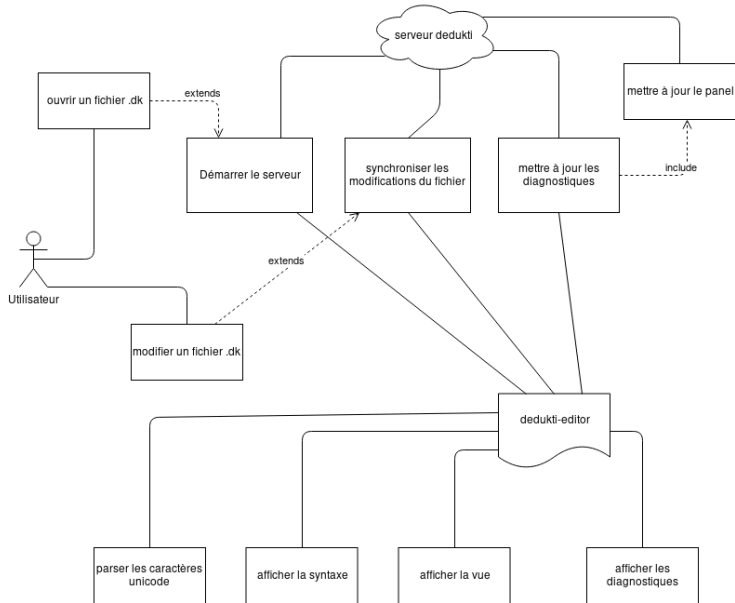
Dépend de la position du curseur

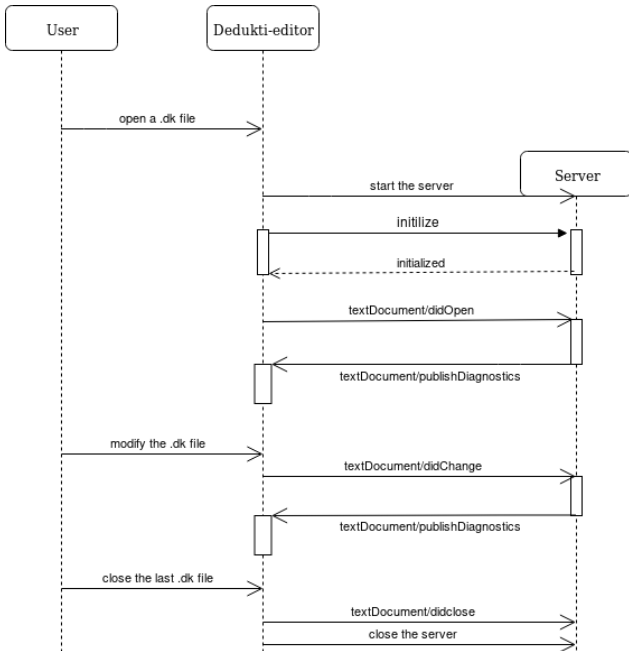


Diagnostics

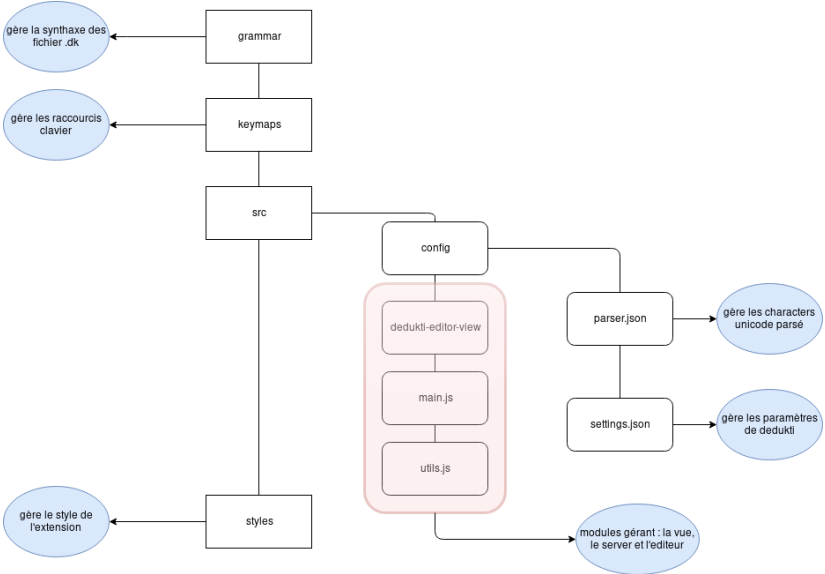
On utilise le message `showDiagnostics` pour afficher le résultat de la vérification de la preuve et réceptionner les données liées à la vue.







Maintenance



Conclusion

Les apports du stage :

- Accroissement des compétences en JavaScript
- L'importance de la *culture informatique* et de la veille technologique
- Une découverte du monde de la recherche

Projet futur

- Rendre l'application compatible avec plus d'assistants de preuve
- Rendre le seveur Dedukti disponible sur d'autres éditeurs de texte en développant d'autres extensions