



Azul
ожон



LIRMM



Une approche déclarative pour la génération de modèles

Adel Ferdjoukh

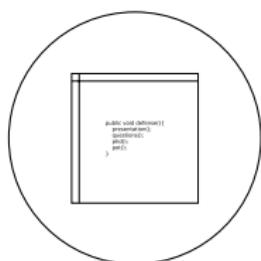
Encadrement: Marianne Huchard, Clémentine Nebut
Eric Bourreau et Annie Chateau

10/2013 - 09/2016

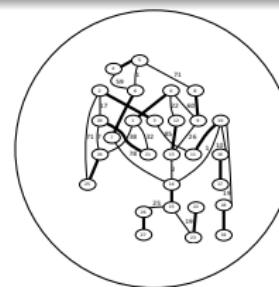
Soutenance de Thèse, le 14 Octobre 2016 à Montpellier

Motivation

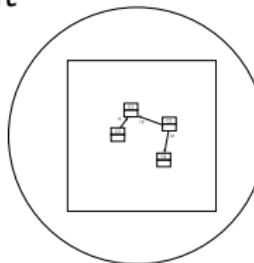
Quel est le point commun entre ?



Langage Nit



Scaffolding

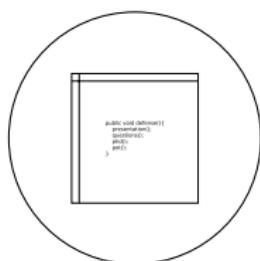


Modèles socio-environnementaux

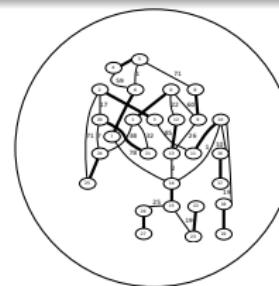
À première vue, aucun.

Motivation

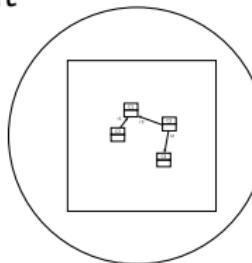
Quel est le point commun entre ?



Langage Nit



Scaffolding



Modèles socio-environnementaux

À première vue, aucun.

Motivation

Deux points communs

- Données représentées sous forme de modèles logiciels à différents niveaux d'abstraction.
- Rareté de ces données pour validation.

Exemple

Tester le compilateur de Nit au moment de son développement

Motivation

Deux points communs

- Données représentées sous forme de modèles logiciels à différents niveaux d'abstraction.
- Rareté de ces données pour validation.

Exemple

Tester le compilateur de Nit au moment de son développement

Motivation

En Ingénierie Dirigée par les Modèles

Un modèle est structuré par un modèle plus abstrait, le métamodèle.

Besoin de beaucoup de modèles

- Savoir si les métamodèles conçus sont valides.
- Tester les programmes les manipulant.

Notre solution

La génération automatique de ces modèles.

Motivation

En Ingénierie Dirigée par les Modèles

Un modèle est structuré par un modèle plus abstrait, le métamodèle.

Besoin de beaucoup de modèles

- Savoir si les métamodèles conçus sont valides.
- Tester les programmes les manipulant.

Notre solution

La génération automatique de ces modèles.

Motivation

En Ingénierie Dirigée par les Modèles

Un modèle est structuré par un modèle plus abstrait, le métamodèle.

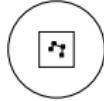
Besoin de beaucoup de modèles

- Savoir si les métamodèles conçus sont valides.
- Tester les programmes les manipulant.

Notre solution

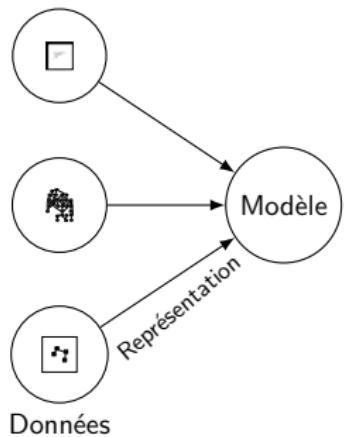
La génération automatique de ces modèles.

Motivation

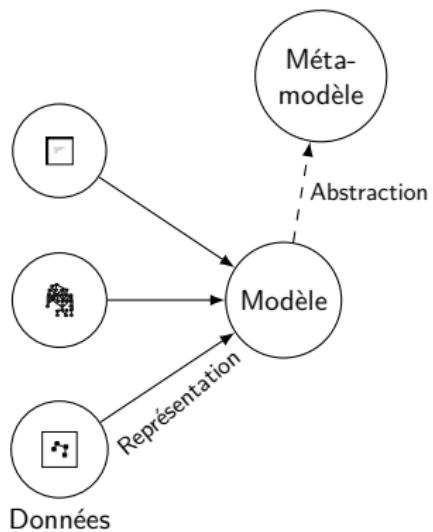


Données

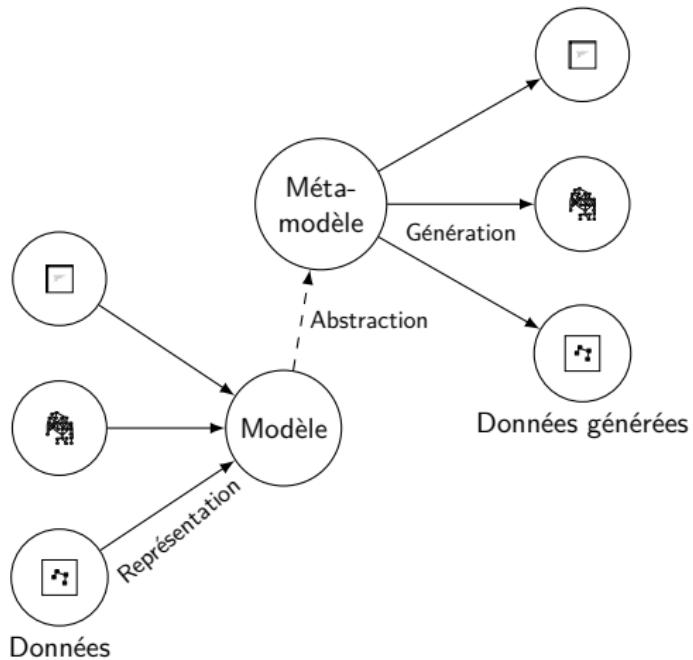
Motivation



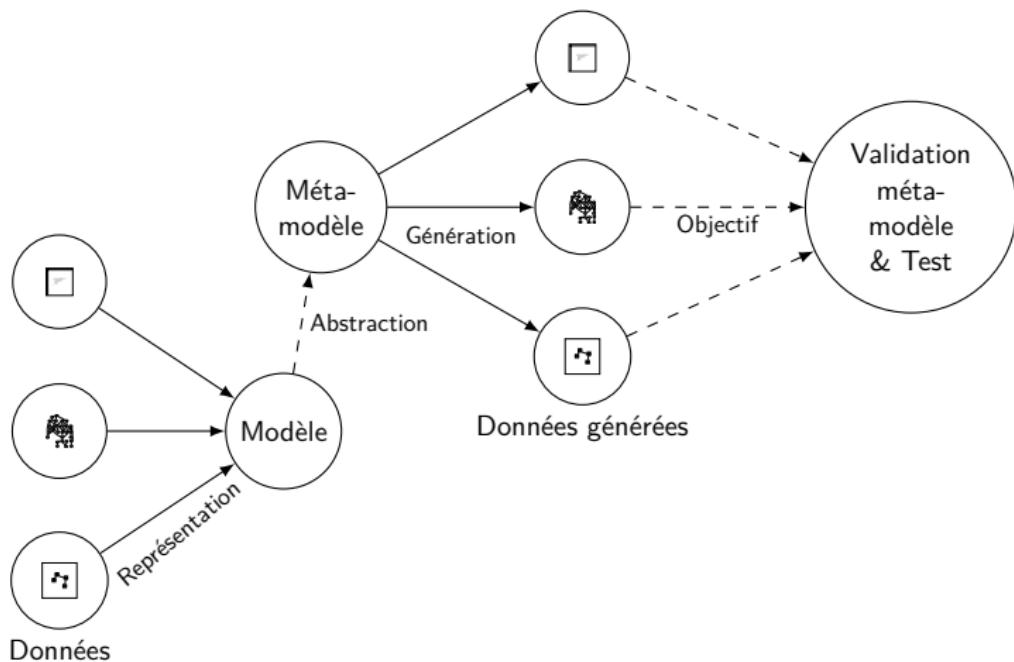
Motivation



Motivation



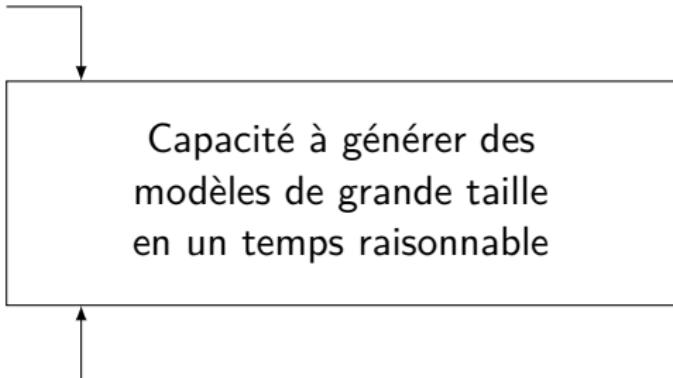
Motivation



Motivation

L'utilisation des modèles permet de dégager des objectifs.

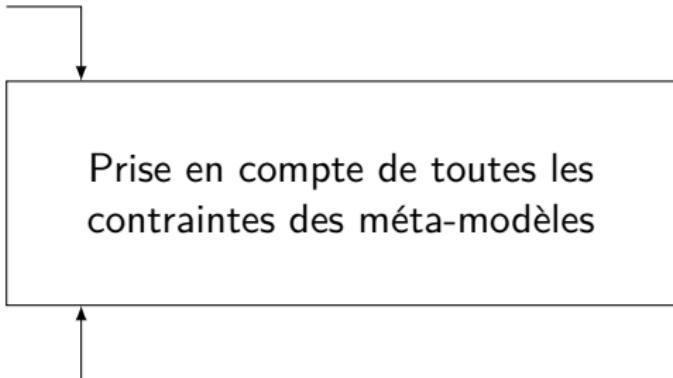
-  Modèle de grande taille
-  Prise en compte des contraintes
-  Vraisemblance
-  Diversité
-  Automatisation



Motivation

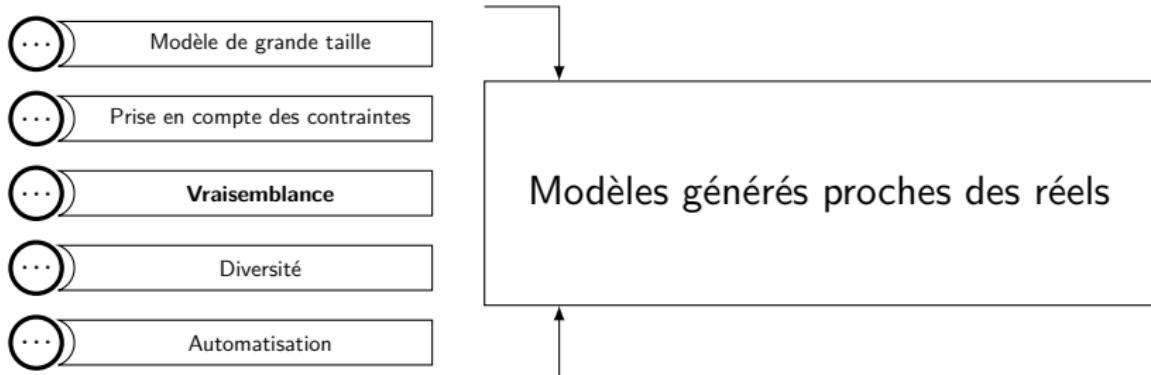
L'utilisation des modèles permet de dégager des objectifs.

-  Modèle de grande taille
-  Prise en compte des contraintes
-  Vraisemblance
-  Diversité
-  Automatisation



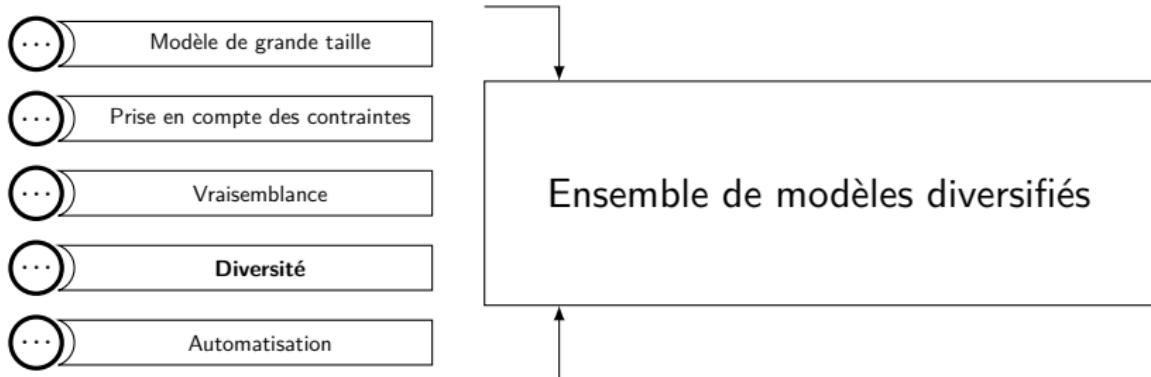
Motivation

L'utilisation des modèles permet de dégager des objectifs.



Motivation

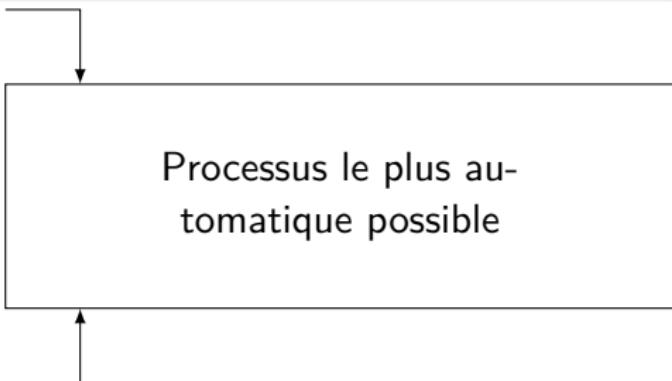
L'utilisation des modèles permet de dégager des objectifs.



Motivation

L'utilisation des modèles permet de dégager des objectifs.

-  Modèle de grande taille
-  Prise en compte des contraintes
-  Vraisemblance
-  Diversité
-  Automatisation



Plan

① Ingénierie Dirigée par les Modèles

② État de l'art

Contributions

③ Modèle de grande taille + OCL

④ Solutions vraisemblables

⑤ Solutions diverses

⑥ Conclusion & Perspectives

Ingénierie dirigée par les modèles

IDM

Paradigme du génie logiciel qui recommande l'utilisation intensive des modèles au cours du développement logiciel.

- Représentation abstraite d'un aspect d'un système.
 - Conception, réalisation et validation de cet aspect.

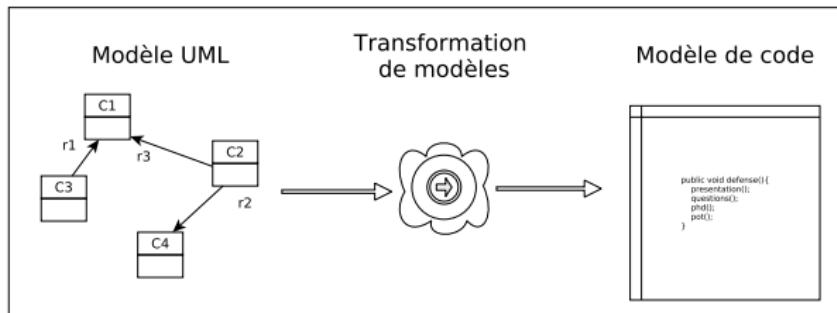
Ingénierie dirigée par les modèles

IDM

Paradigme du génie logiciel qui recommande l'utilisation intensive des modèles au cours du développement logiciel.

- Représentation abstraite d'un aspect d'un système.
 - Conception, réalisation et validation de cet aspect.

Exemple de processus dirigé modèle: génération de code



Ingénierie dirigée par les modèles

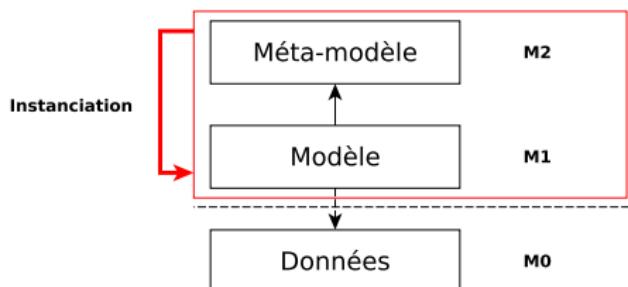
Hiérarchie des modèles

En IDM, un modèle est défini par un modèle plus abstrait, le métamodèle.

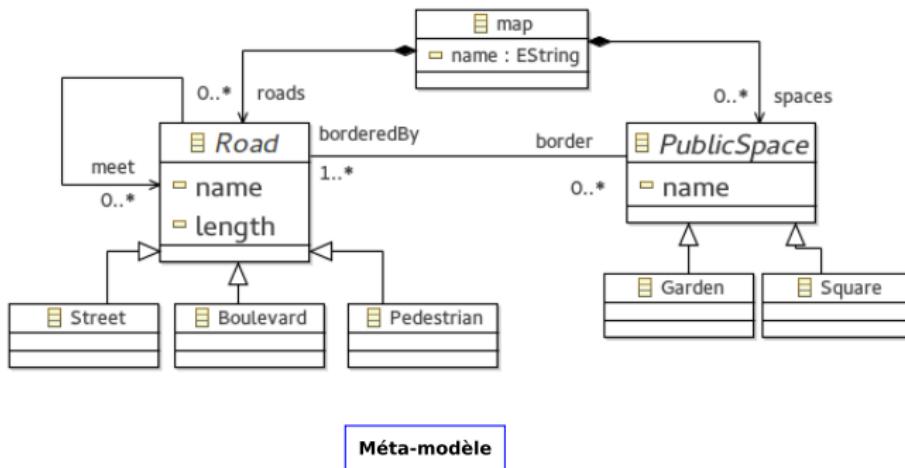
Ingénierie dirigée par les modèles

Hiérarchie des modèles

En IDM, un modèle est défini par un modèle plus abstrait, le métamodèle.



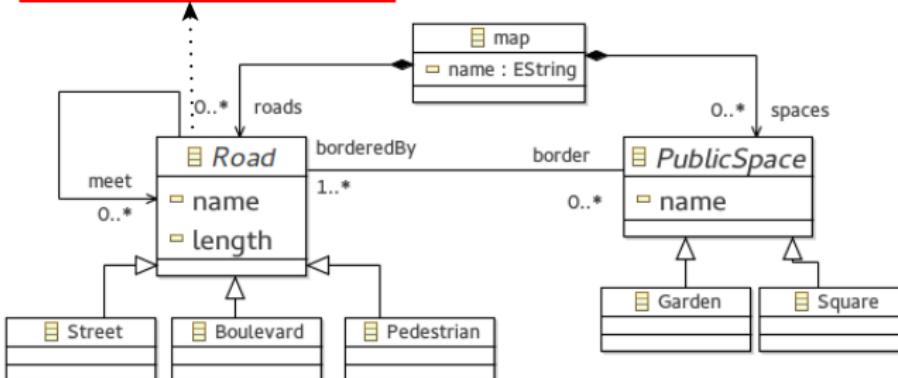
Méta-modèle et OCL



Méta-modèle et OCL

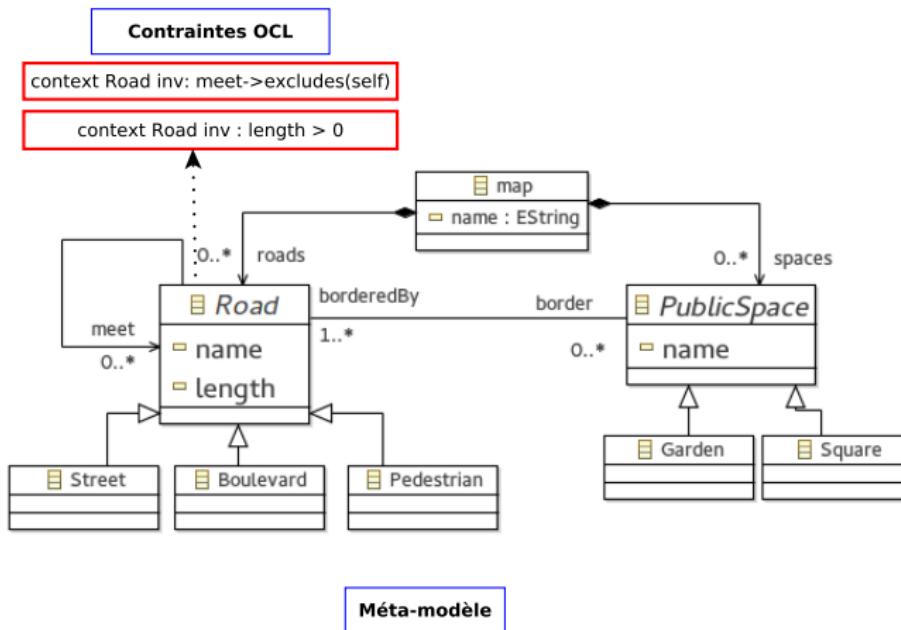
Contraintes OCL

context Road inv : length > 0

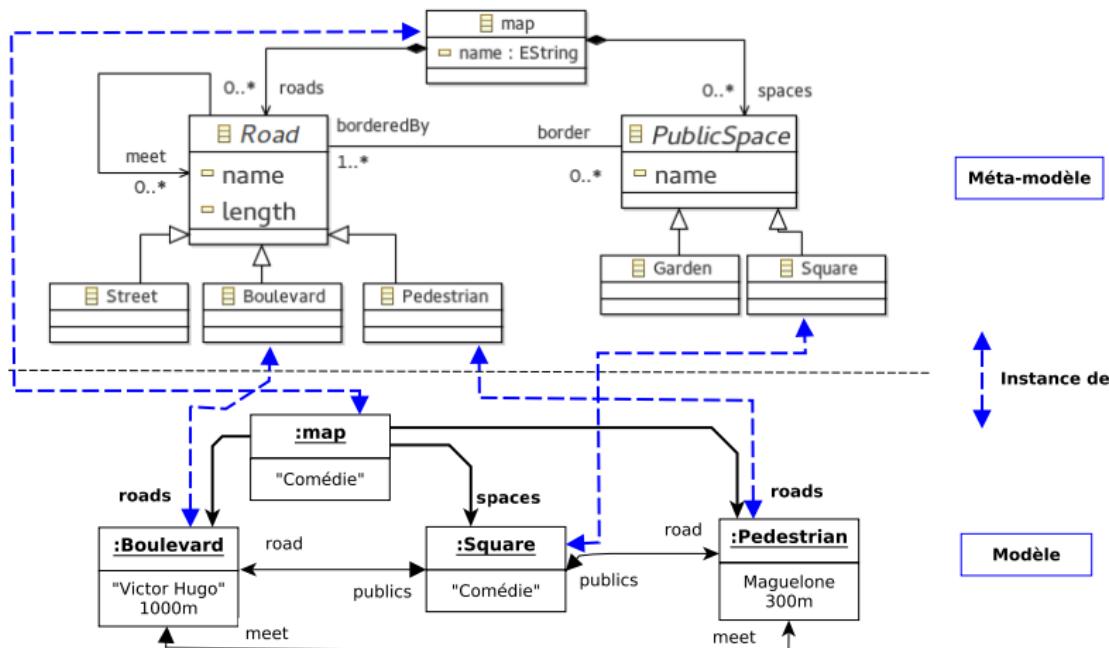


Méta-modèle

Méta-modèle et OCL



Méta-modèle et Modèle conforme



Plan

- ① Ingénierie Dirigée par les Modèles
- ② État de l'art

Contributions

- ③ Modèle de grande taille + OCL
- ④ Solutions vraisemblables
- ⑤ Solutions diverses
- ⑥ Conclusion & Perspectives

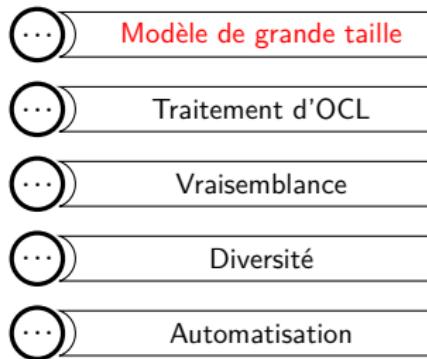
Approches de génération de modèles

| Auteurs | Paradigme | Critères de comparaison | | | | |
|--------------|---------------------|-------------------------|-----|---------|------------|-----------|
| | | Auto | OCL | Échelle | Pertinence | Diversité |
| Mougenot, 09 | Arbre aléatoire | + | - | + | + | - |
| Ehrig, 06 | Grammaire de graphe | - | - | + | - | - |
| Wu, 12 | SMT | + | + | - | + | - |
| Cabot, 08 | CSP | + | + | - | - | - |
| Sen, 08 | Alloy | - | + | - | - | - |
| Brottier, 06 | Fragments | + | - | ? | - | + |
| Cadavid, 12 | Recuit simulé | + | - | ? | - | + |

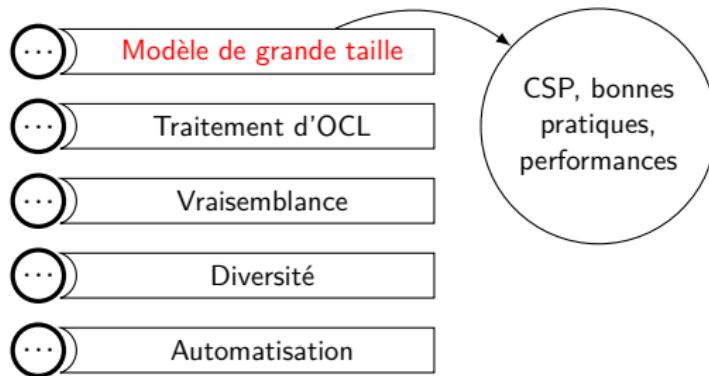
Approches de génération de modèles

| Auteurs | Paradigme | Critères de comparaison | | | | |
|--------------|---------------------|-------------------------|-----|---------|------------|-----------|
| | | Auto | OCL | Échelle | Pertinence | Diversité |
| Mougenot, 09 | Arbre aléatoire | + | - | + | + | - |
| Ehrig, 06 | Grammaire de graphe | - | - | + | - | - |
| Wu, 12 | SMT | + | + | - | + | - |
| Cabot, 08 | CSP | + | + | - | - | - |
| Sen, 08 | Alloy | - | + | - | - | - |
| Brottier, 06 | Fragments | + | - | ? | - | + |
| Cadavid, 12 | Recuit simulé | + | - | ? | - | + |
| GRIMM | CSP, ... | + | + | + | + | + |

Progression



Progression



Génération de modèles

Paradigme choisi: CSP

Programmation par contraintes - CSP

Paradigme issu de l'intelligence artificielle utilisé pour résoudre des problèmes combinatoires et de décision (> 1980).

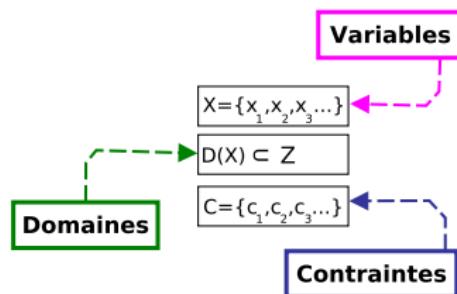
Génération de modèles

Paradigme choisi: CSP

Programmation par contraintes - CSP

Paradigme issu de l'intelligence artificielle utilisé pour résoudre des problèmes combinatoires et de décision (> 1980).

Programmation par contraintes

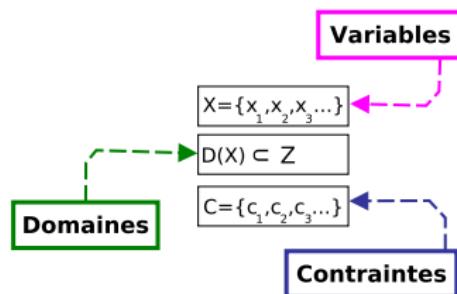


Solution du CSP

Affectation de valeurs à toutes les variables respectant toutes les contraintes.

Les solutions sont trouvées par un solveur de contraintes.

Programmation par contraintes



Solution du CSP

Affectation de valeurs à toutes les variables respectant toutes les contraintes.

Les solutions sont trouvées par un solveur de contraintes.

Programmation par contraintes

- Contrainte: expression à valeur booléenne portant sur un ensemble de variables.
- Exemple: $x_1 > x_2$, $x_1 + x_2 = 5$, ...

Les contraintes globales

- Contrainte portant sur n variables (n-aire).
- Modélisation facilitée et CSP compact.
- Exemple: Alldiff qui rend n variables toutes différentes.
- Algorithmes de résolution (filtrage) dédiés.

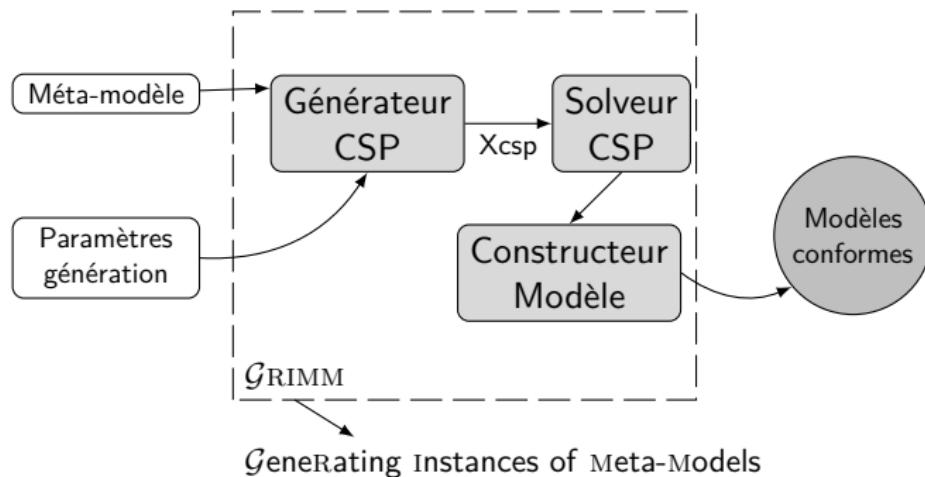
Programmation par contraintes

- Contrainte: expression à valeur booléenne portant sur un ensemble de variables.
- Exemple: $x_1 > x_2$, $x_1 + x_2 = 5$, ...

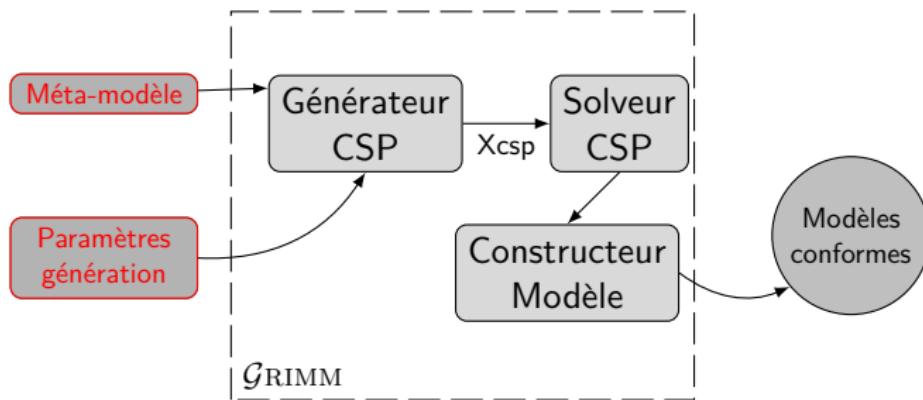
Les contraintes globales

- Contrainte portant sur n variables (n -aire).
- Modélisation facilitée et CSP compact.
- Exemple: Alldiff qui rend n variables toutes différentes.
- Algorithmes de résolution (filtrage) dédiés.

Processus de génération de modèles

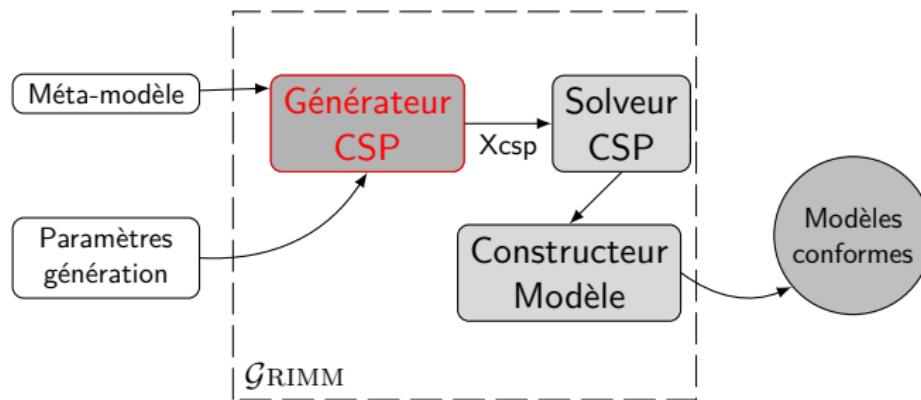


Processus de génération de modèles



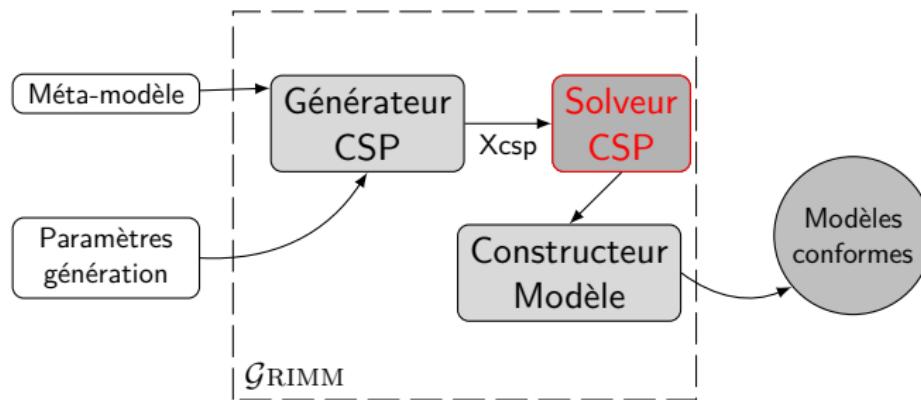
Generating Instances of Meta-Models

Processus de génération de modèles



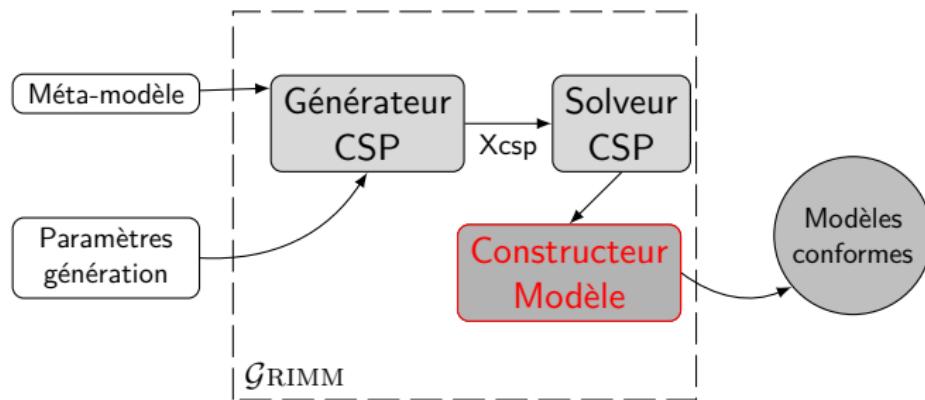
Generating Instances of Meta-Models

Processus de génération de modèles



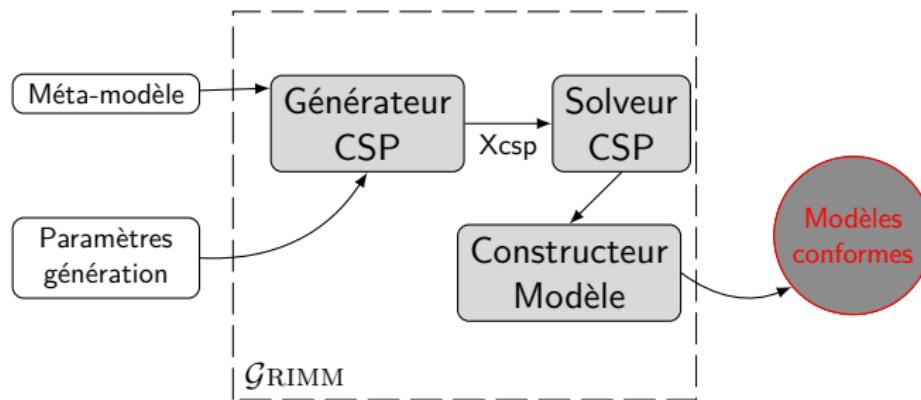
Generating Instances of Meta-Models

Processus de génération de modèles



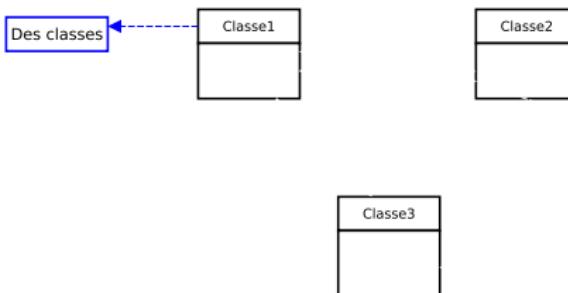
Generating Instances of Meta-Models

Processus de génération de modèles



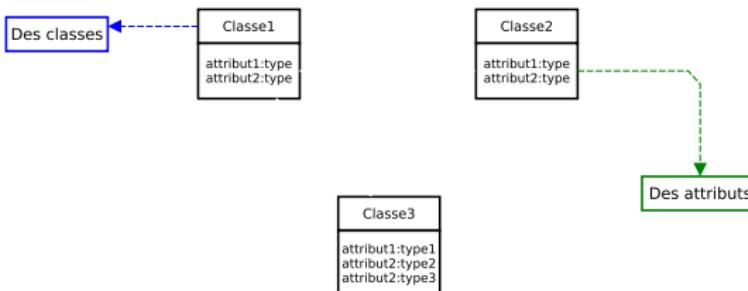
Generating Instances of Meta-Models

Parties du métamodèle à formaliser



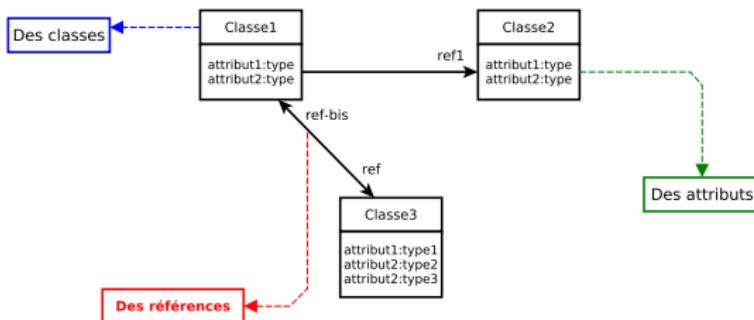
- Créer des instances pour les classes concrètes et l'héritage.

Parties du métamodèle à formaliser



- Créer des instances pour les classes concrètes et l'héritage.
- Supporter tout type d'attributs.

Parties du métamodèle à formaliser



- Créer des instances pour les classes concrètes et l'héritage.
- Supporter tout type d'attributs.
- Gérer les références dans les deux directions.

Méta-modèle vers CSP

- Création de variables, domaines et contraintes pour représenter les instances des classes, les attributs et les références.

Bonnes pratiques de modélisation en CSP

- Minimiser le nombre de variables.
- Compackter les domaines.
- Minimiser le nombre de contraintes.
- Utiliser les contraintes globales.

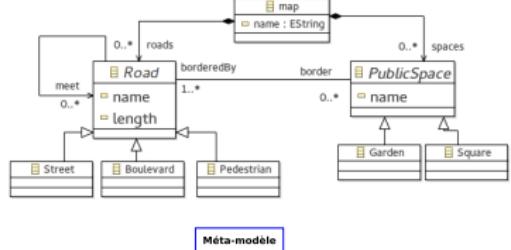
Méta-modèle vers CSP

- Création de variables, domaines et contraintes pour représenter les instances des classes, les attributs et les références.

Bonnes pratiques de modélisation en CSP

- Minimiser le nombre de variables.
- Compackter les domaines.
- Minimiser le nombre de contraintes.
- Utiliser les contraintes globales.

Configuration

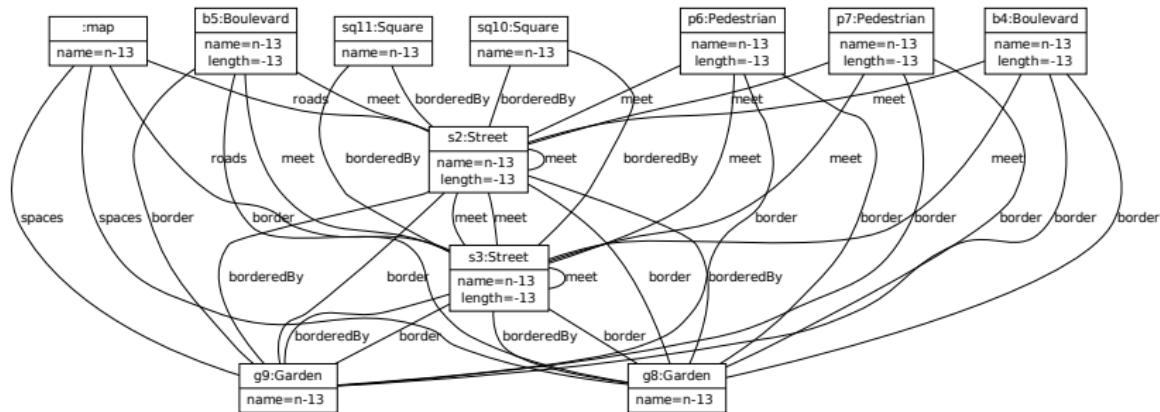


Paramètres de configuration

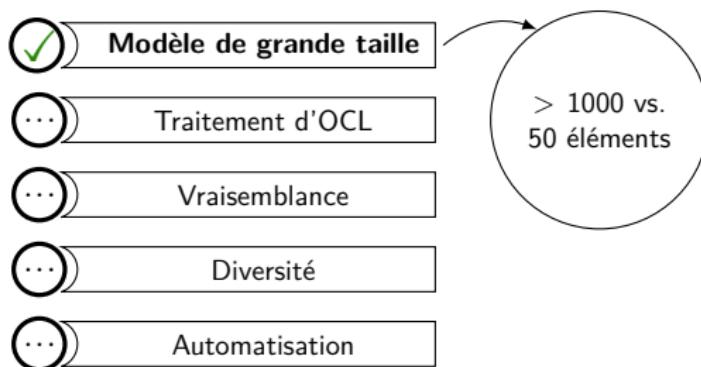
% -----
 Number of instances per Class
 % -----
 Street=2
 Boulevard=2
 Pedestrian=2
 Garden=2
 Square=2

% -----
 Some others
 % -----
 RefsBound=3
 AttributesBound=2

Modèle produit

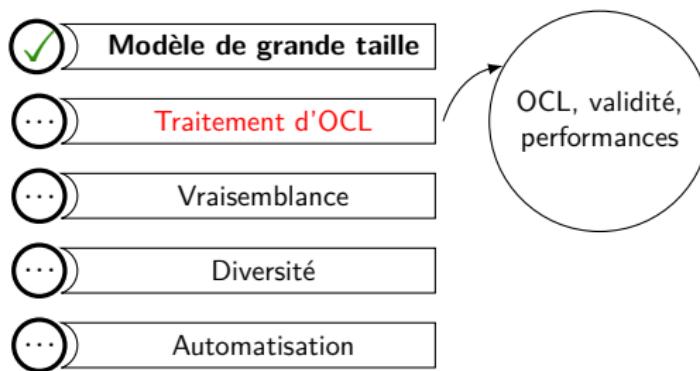


Progression

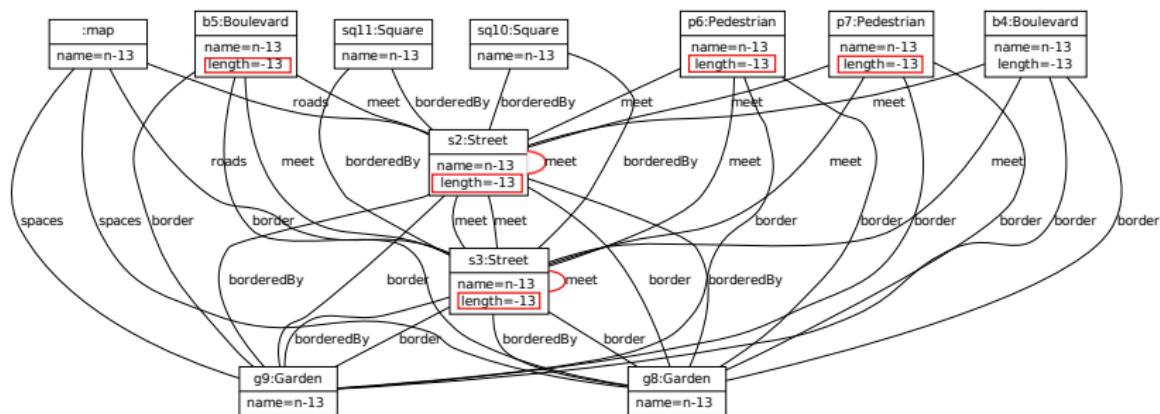


Ferdjoukh et al., ICTAI, 2013.

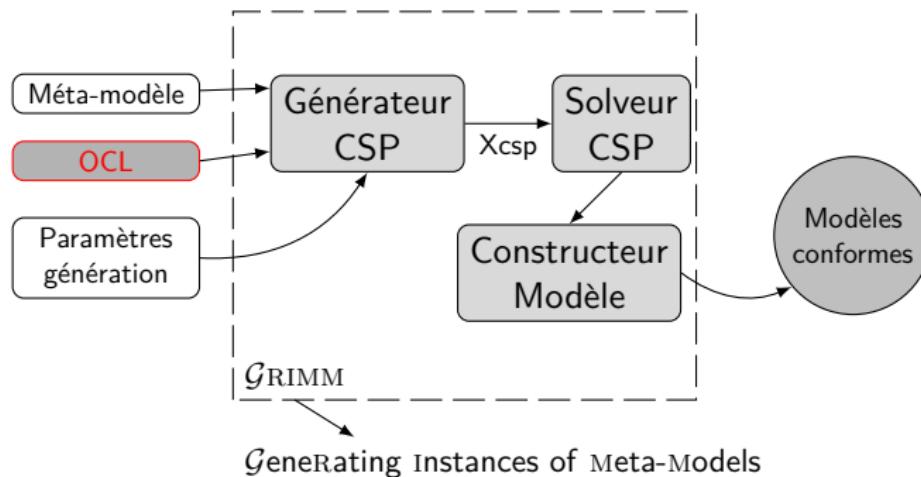
Progression



Modèle produit



OCL vers CSP



OCL vers CSP

- Traitement des contraintes OCL simples, navigation de références, opération sur les collections.

Bonnes pratiques de modélisation en CSP

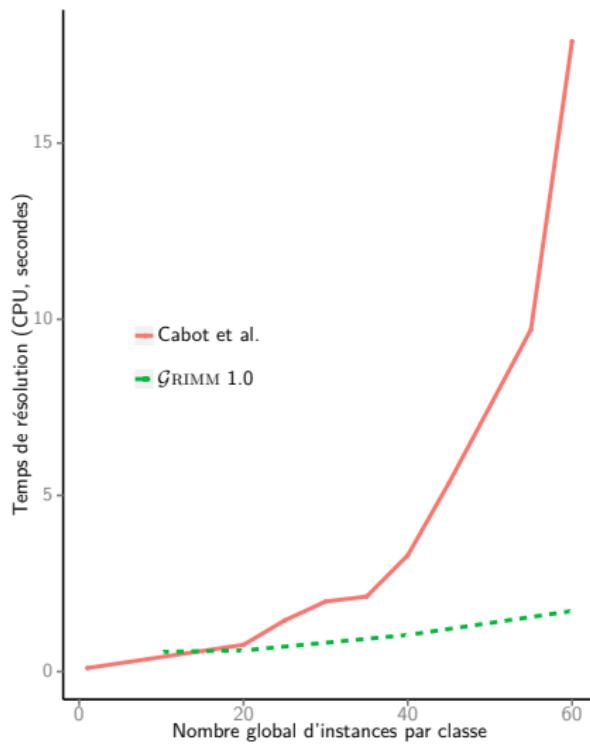
- Ne créer de nouvelles variables que très rarement.
- Éviter les contraintes quand c'est possible.
- Pré-traiter au maximum.
- Utiliser les contraintes globales.

Expérimentations

Objectifs des expérimentations

- ① Sommes-nous plus efficaces que les approches existantes ?
- ② Notre approche passe-t-elle bien à l'échelle ?
- ③ Quel est l'impact d'OCL sur les performances ?

Comparaison avec une autre approche



Modèle de grande taille

Protocole

- 15 méta-modèles de différentes tailles.
- des modèles entre 50 et 1000 éléments.
- Calcul du temps de résolution.

Résultats

- Globalement rapide (quelques secondes pour 1000 instances de classe).
- Entre 0.58 et 11 secondes pour 12 méta-modèles.
- 18s et 28s pour 3 méta-modèles.

Modèle de grande taille

Protocole

- 15 méta-modèles de différentes tailles.
- des modèles entre 50 et 1000 éléments.
- Calcul du temps de résolution.

Résultats

- Globalement rapide (quelques secondes pour 1000 instances de classe).
- Entre 0.58 et 11 secondes pour 12 méta-modèles.
- 18s et 28s pour 3 méta-modèles.

Prise en compte d'OCL

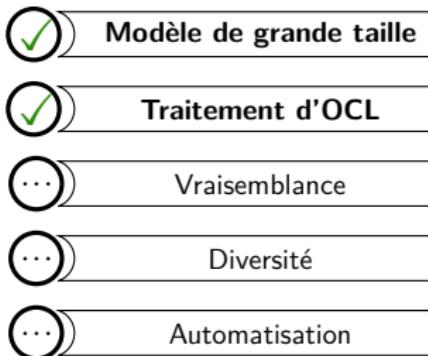
Protocole

- Contraintes OCL pour 4 métamodèles.
- Génération avec ou sans OCL.
- Comparaison du temps de résolution.

Prise en compte d'OCL

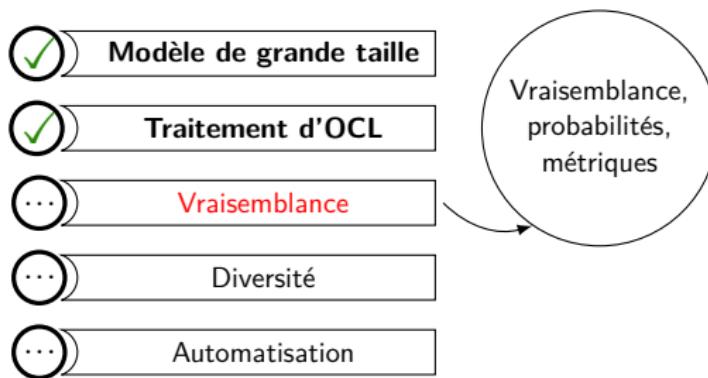
| méta-modèle | Petri nets | | ER | | Feature | | Graph Colour | |
|-------------|-------------------------|------|-------|------|---------|------|--------------|------|
| #instances | OCL ? | | OCL ? | | OCL ? | | OCL ? | |
| | oui | non | oui | non | oui | non | oui | non |
| | Temps de résolution (s) | | | | | | | |
| | 50 | 0.46 | 0.47 | 0.58 | 0.56 | 0.31 | 0.31 | 0.36 |
| 100 | 0.70 | 0.69 | 0.67 | 0.60 | 0.36 | 0.35 | 0.43 | 0.39 |
| 300 | 2.26 | 2.25 | 1.88 | 1.72 | 0.42 | 0.41 | 0.62 | 0.47 |
| 600 | 2.57 | 2.58 | 16.1 | 15.5 | 0.47 | 0.49 | 0.85 | 0.60 |

Progression

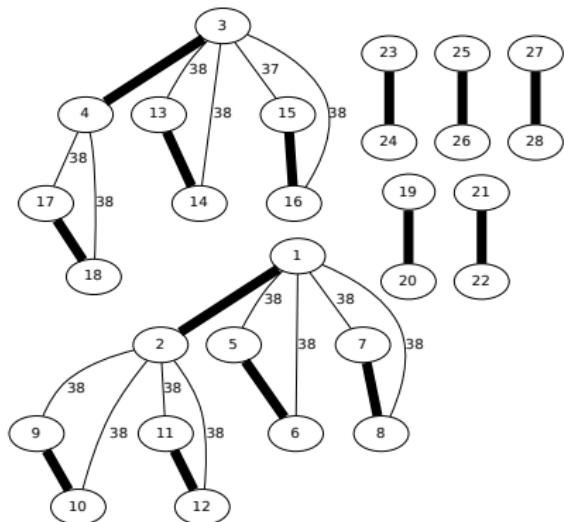


Ferdjoukh et al., Modelsward, 2015.

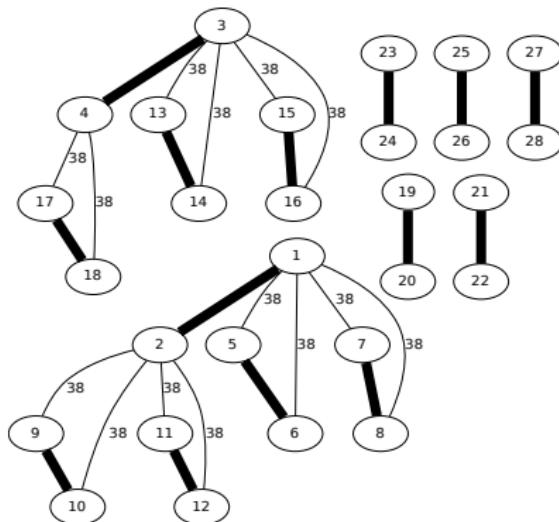
Progression



Problématique

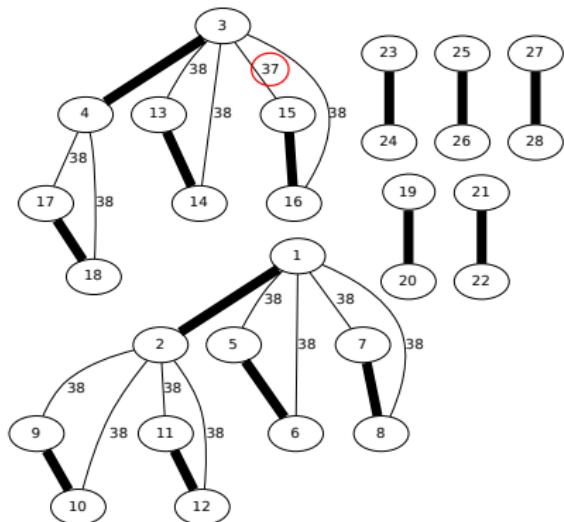


Graphe 1

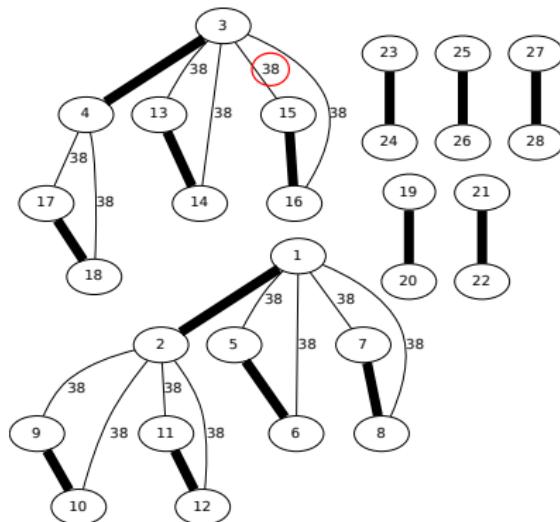


Graphe 2

Problématique



Graphe 1



Graphe 2

Objectif

Constat

Nos modèles sont peu réalistes et se ressemblent beaucoup.

Objectif

Améliorer leur diversité intra-modèle et leur vraisemblance.

Objectif

Constat

Nos modèles sont peu réalistes et se ressemblent beaucoup.

Objectif

Améliorer leur diversité intra-modèle et leur vraisemblance.

Comment améliorer la vraisemblance

Solution envisagée

- Aller plus loin dans l'arbre des solutions.
- Injecter de l'aléatoire dans le solveur (pour un processus moins prévisible.)
- Injecter de l'aléatoire en dehors du solveur.

Critique/Analyse

- Même la 1^{ère} et la 100.000^{ème} sont proches.
- Ralentissements des performances et résultats peu probants.
- Solution retenue.

Comment améliorer la vraisemblance

Solution envisagée

- Aller plus loin dans l'arbre des solutions.
- Injecter de l'aléatoire dans le solveur (pour un processus moins prévisible.)
- Injecter de l'aléatoire en dehors du solveur.

Critique/Analyse

- Même la 1^{ère} et la 100.000^{ème} sont proches.
- Ralentissements des performances et résultats peu probants.
- Solution retenue.

Comment améliorer la vraisemblance

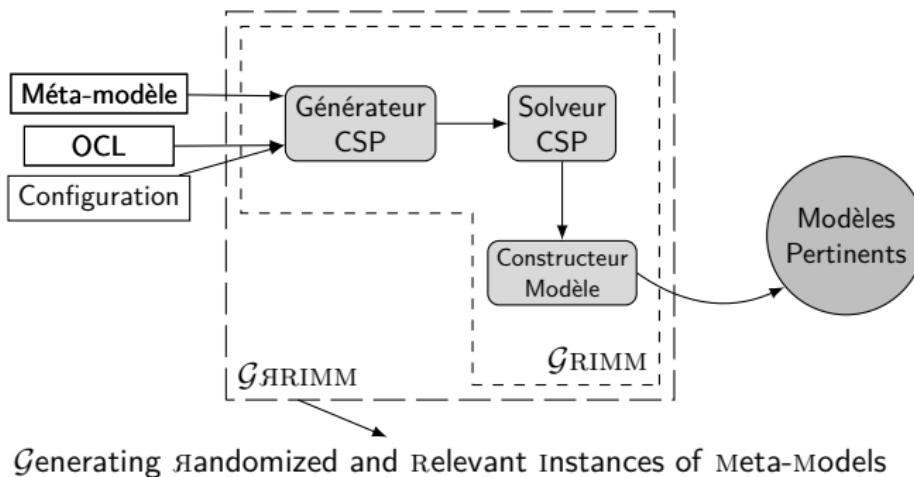
Solution envisagée

- Aller plus loin dans l'arbre des solutions.
- Injecter de l'aléatoire dans le solveur (pour un processus moins prévisible.)
- Injecter de l'aléatoire en dehors du solveur.

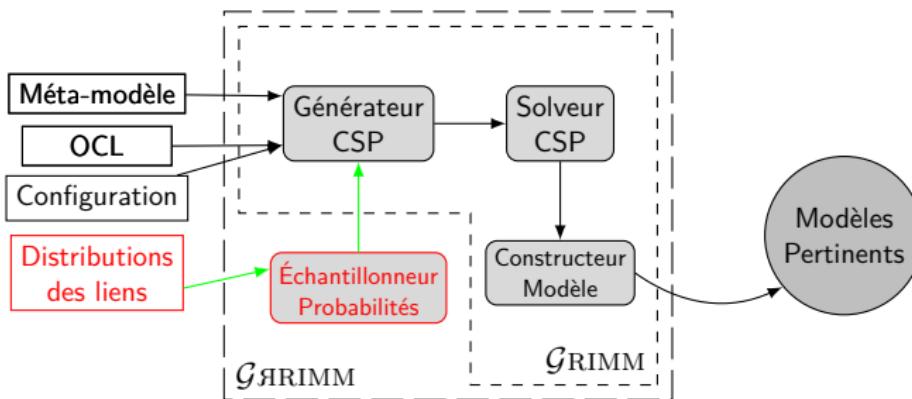
Critique/Analyse

- Même la 1^{ère} et la 100.000^{ème} sont proches.
- Ralentissements des performances et résultats peu probants.
- Solution retenue.

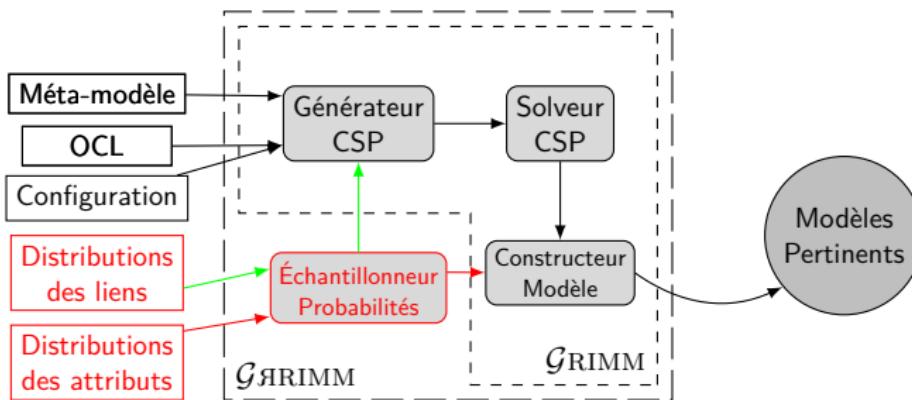
Génération d'instances réalistes



Génération d'instances réalistes



Génération d'instances réalistes



Étude de cas: programmes Java

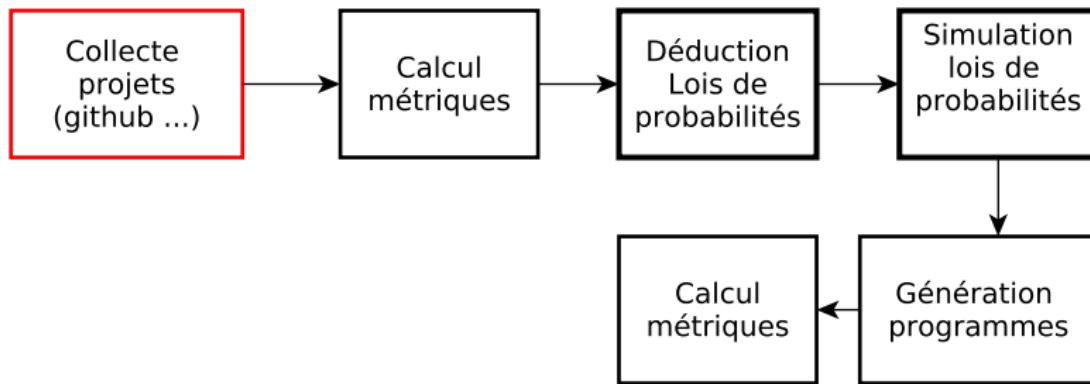
Étude de cas

Génération de programmes orientés objets (exemple Java).

Objectifs

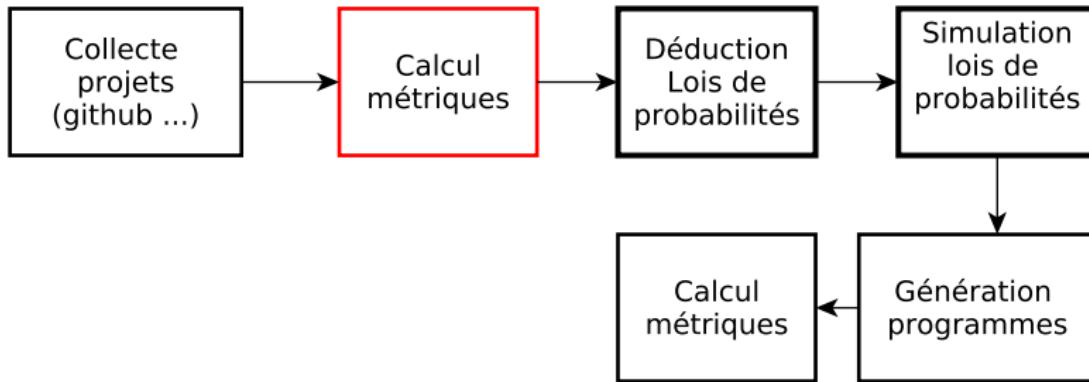
- Des modèles conformes aux souhaits de l'utilisateur.
- Des modèles générés proches des modèles réels.

Protocole



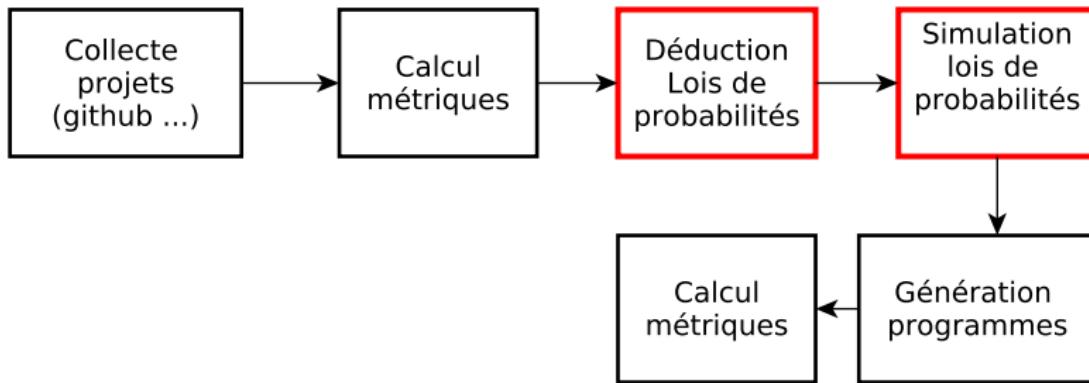
- 200 projets java analysés de 20 à 9000 classes.

Protocole



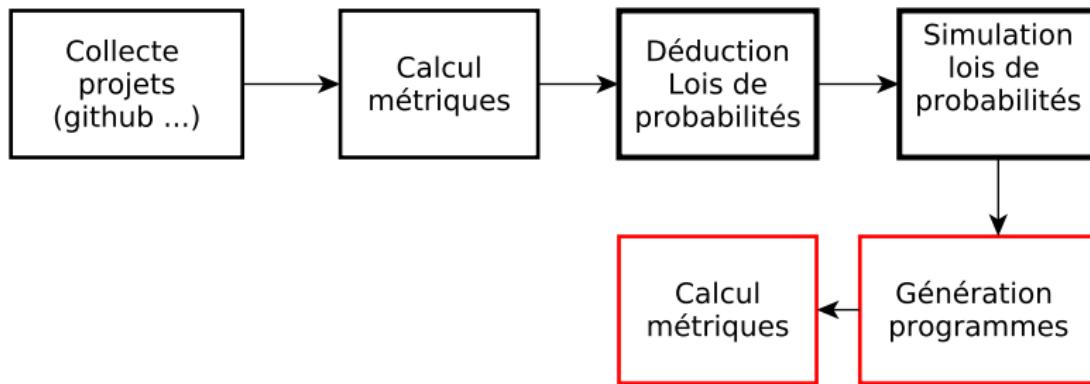
- Métriques choisies: attributs par classe, constructeurs par classe, visibilité, ...

Protocole



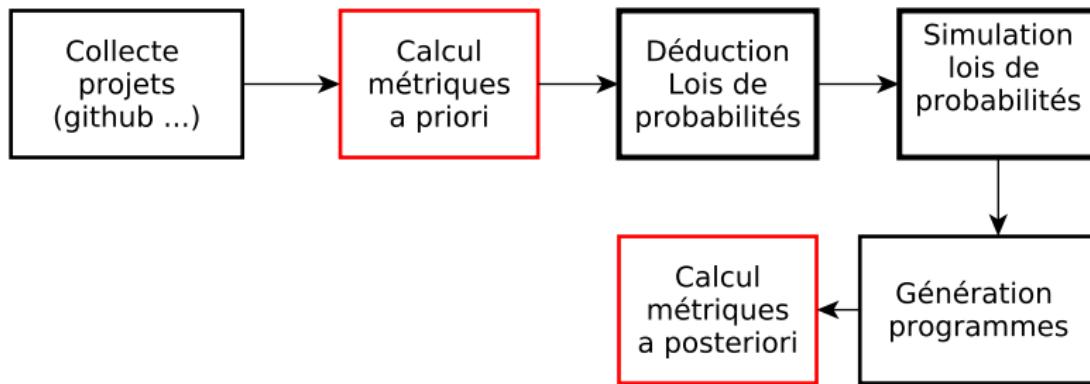
- Les ajouter au générateur de CSP ou bien à l'étape de construction de modèles

Protocole



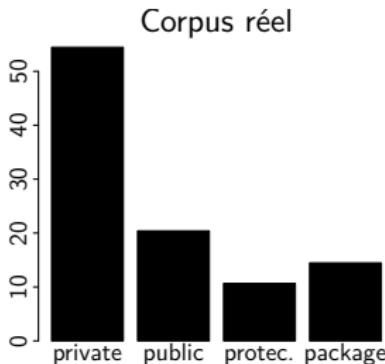
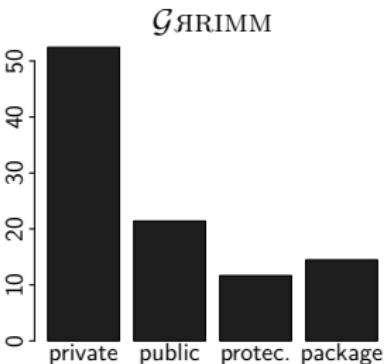
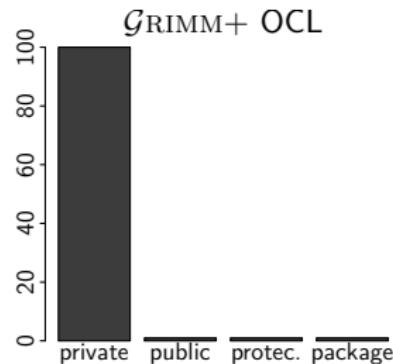
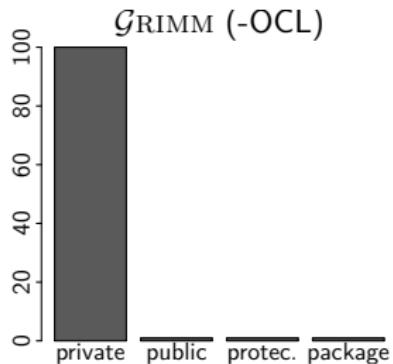
- Générer des nouveaux projets Java.

Protocole



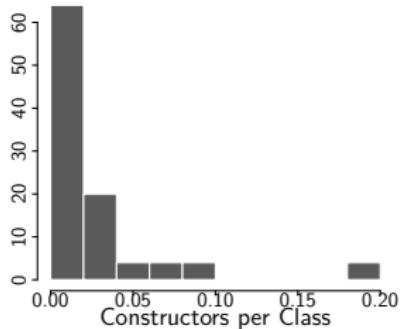
- Comparer les métriques des modèles générés aux réelles.

Résultats: visibilité des attributs

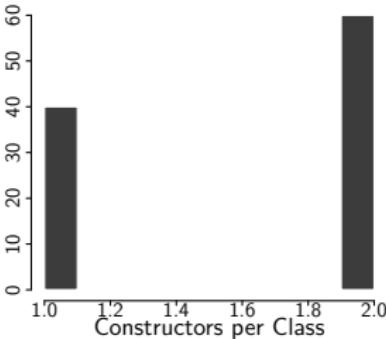


Résultats: constructeurs par classe

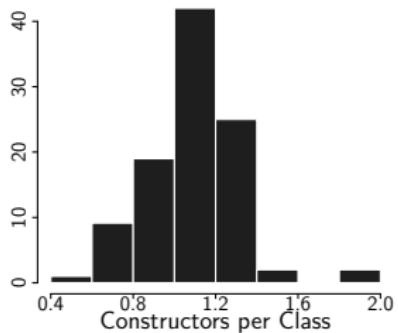
\mathcal{G} RIMM (-OCL)



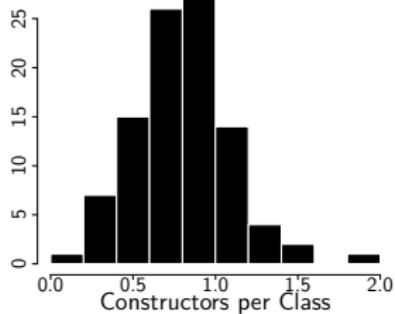
\mathcal{G} RIMM+OCL



\mathcal{G} RIMM



Corpus réel



Étude de cas: graphes de Scaffold

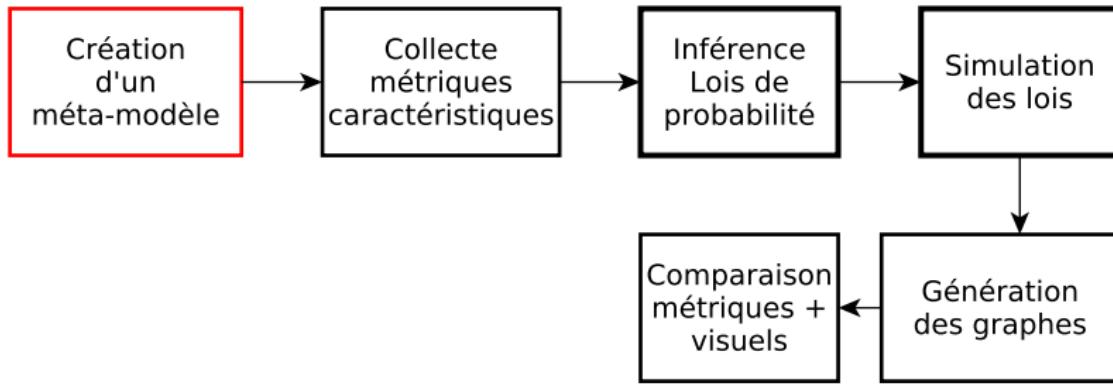
Étude de cas

Génération de graphes de Scaffold utilisés en bio-informatique au cours de l'assemblage des génomes.

Objectifs

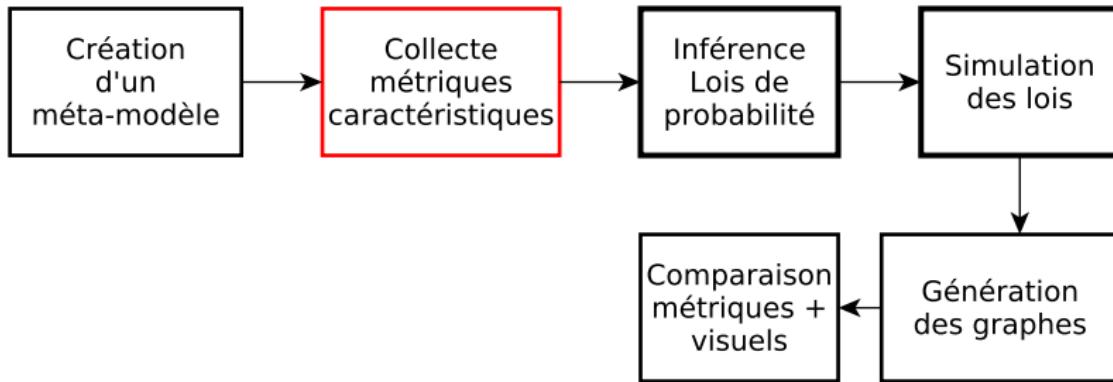
- Générer de grandes quantités de graphes.
- Utiliser les métriques d'un domaine pour améliorer la vraisemblance.

Protocole



- Conception d'un méta-modèle pour les graphes de Scaffold.

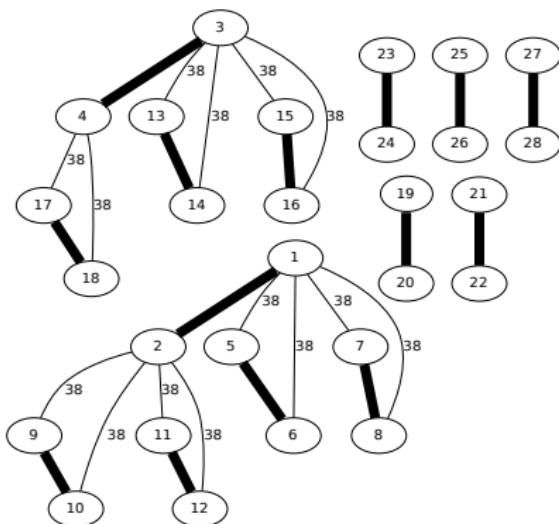
Protocole



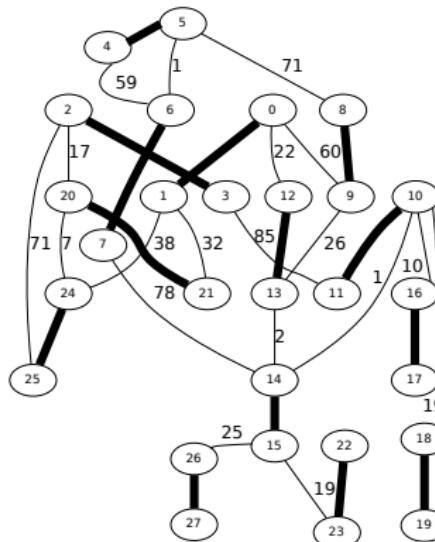
- Métriques choisies: degré des nœuds, répartition des poids des arêtes, connectivité du graphe.

Résultats: graphes de Scaffold

GRIMM

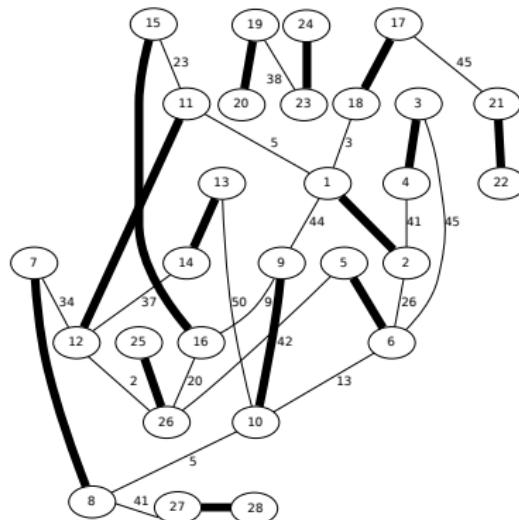


Graphe réel

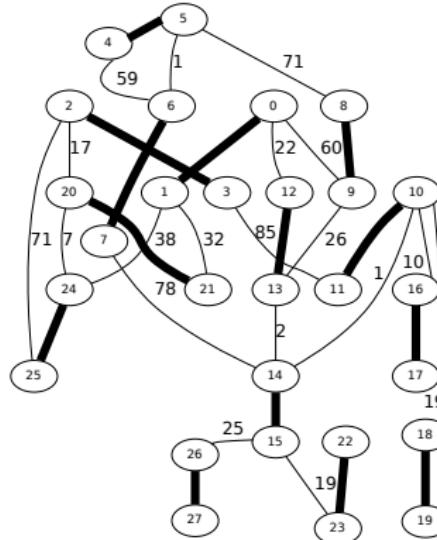


Résultats: graphes de Scaffold

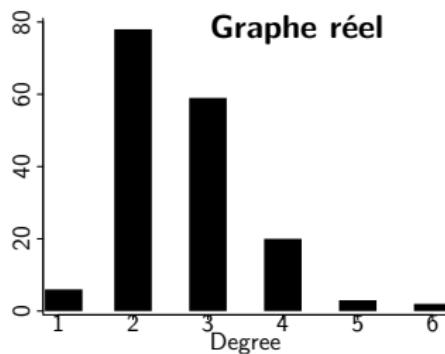
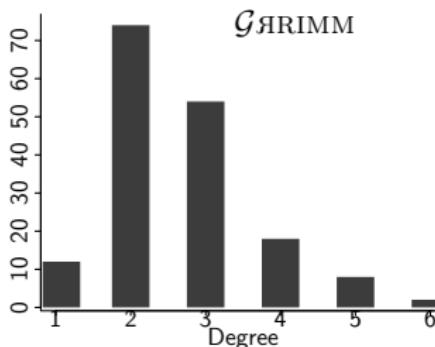
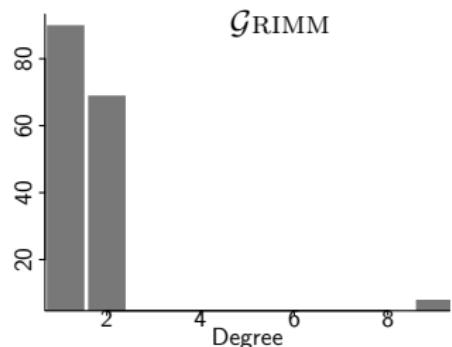
$\mathcal{G}_{\text{RIMM}}$



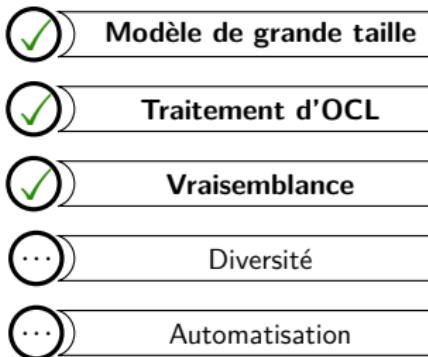
Graphe réel



Résultats: graphes de Scaffold

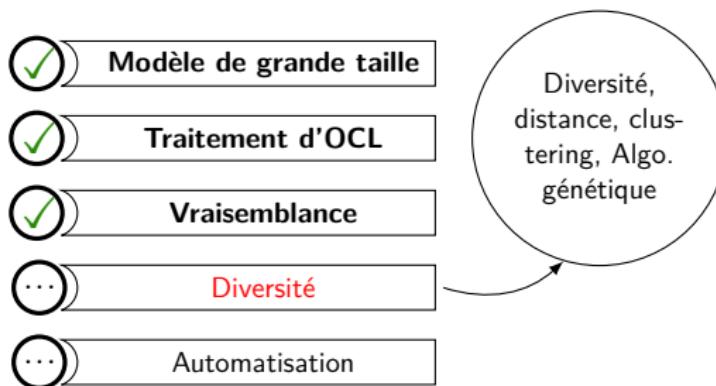


Progression



Ferdjoukh et al., Seke, 2016.

Progression



Génération d'instances diverses

Objectif

Générer des ensembles de modèles différents et diversifiés.

Procédons par étapes

- ➊ Quantifier la différence entre deux modèles.
- ➋ Quantifier et illustrer la diversité d'un ensemble de modèles.
- ➌ Augmenter la diversité d'un ensemble de modèles.

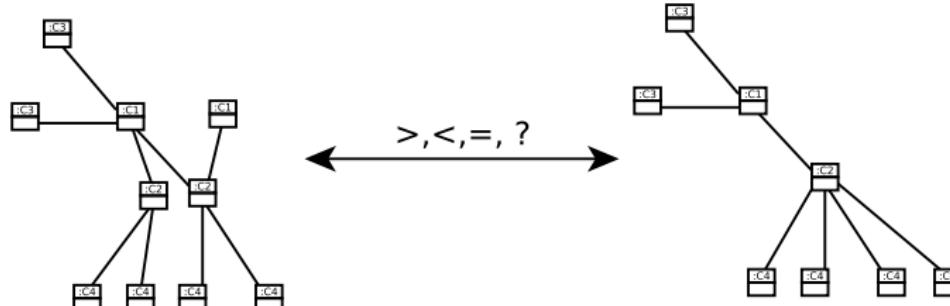
Génération d'instances diverses

Objectif

Générer des ensembles de modèles différents et diversifiés.

Procédons par étapes

- ① Quantifier la différence entre deux modèles.
- ② Quantifier et illustrer la diversité d'un ensemble de modèles.
- ③ Augmenter la diversité d'un ensemble de modèles.



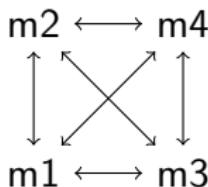
Génération d'instances diverses

Objectif

Générer des ensembles de modèles différents et diversifiés.

Procédons par étapes

- ① Quantifier la différence entre deux modèles.
- ② Quantifier et illustrer la diversité d'un ensemble de modèles.
- ③ Augmenter la diversité d'un ensemble de modèles.



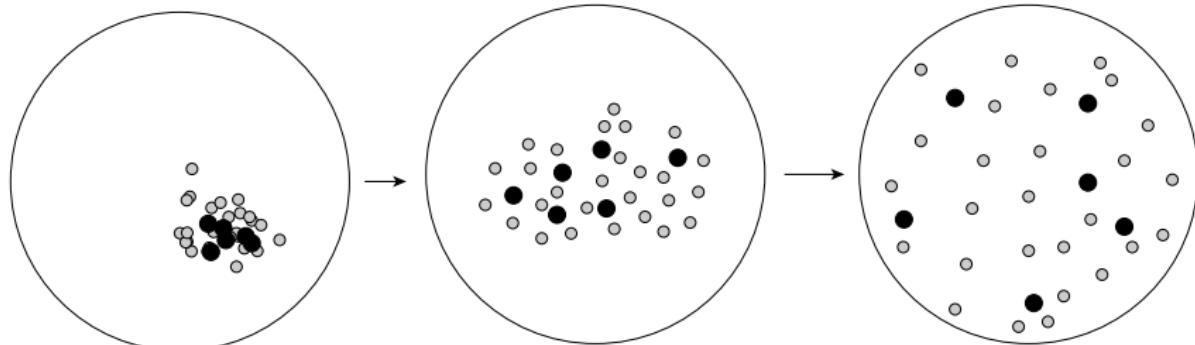
Génération d'instances diverses

Objectif

Générer des ensembles de modèles différents et diversifiés.

Procédons par étapes

- ① Quantifier la différence entre deux modèles.
- ② Quantifier et illustrer la diversité d'un ensemble de modèles.
- ③ Augmenter la diversité d'un ensemble de modèles.

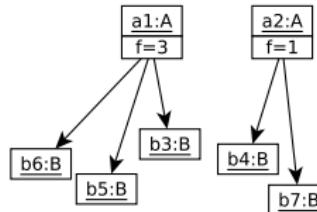
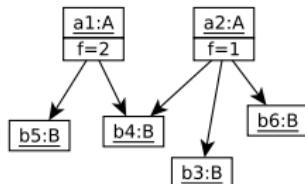


Distances entre modèles

Trois distances

- Basée sur Hamming.
- Basée sur la centralité.
- Basée sur l'édition d'arbres.

Distance de Hamming



$$a = (\underbrace{5, 4, 0,}_{\text{liens}} \underbrace{2,}_{\text{attributs}} \underbrace{4, 3, 6,}_{\text{liens}} \underbrace{1}_{\text{attributs}})$$

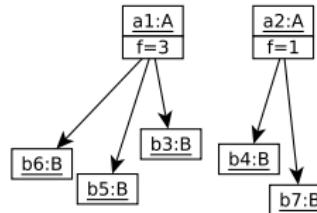
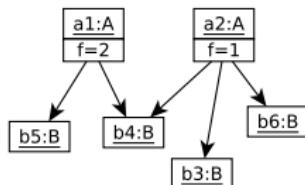
instance a1 *instance a2*

$$b = (\underbrace{6, 5, 3,}_{\text{liens}} \underbrace{3,}_{\text{attributs}} \underbrace{4, 7, 0,}_{\text{liens}} \underbrace{1}_{\text{attributs}})$$

instance a1 *instance a2*

$$\begin{aligned}
 a &= (5, 4, 0, 2, 4, 3, 6, 1) \\
 b &= (6, 5, 3, 3, 4, 7, 0, 1) \\
 d(a, b) &= \frac{1+1+1+1+0+1+1+0}{8} \\
 &= \frac{6}{8}
 \end{aligned}$$

Distance de Hamming



$$a = (\underbrace{5, 4, 0,}_{\text{liens}} \underbrace{2,}_{\text{attributs}} \underbrace{4, 3, 6,}_{\text{liens}} \underbrace{1}_{\text{attributs}})$$

$$b = (\underbrace{6, 5, 3,}_{\text{liens}} \underbrace{3,}_{\text{attributs}} \underbrace{4, 7, 0,}_{\text{liens}} \underbrace{1}_{\text{attributs}})$$

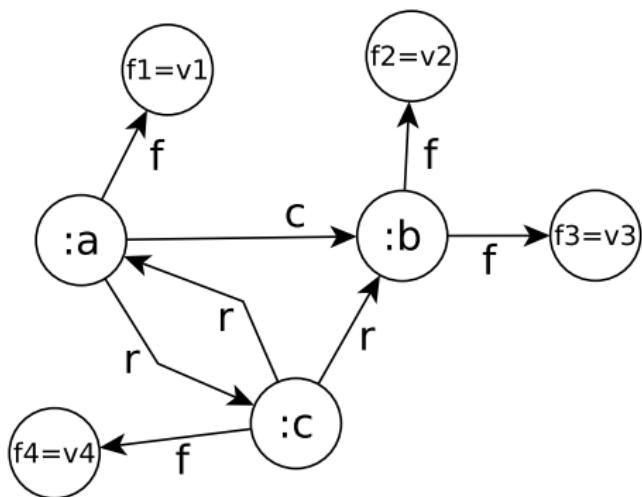
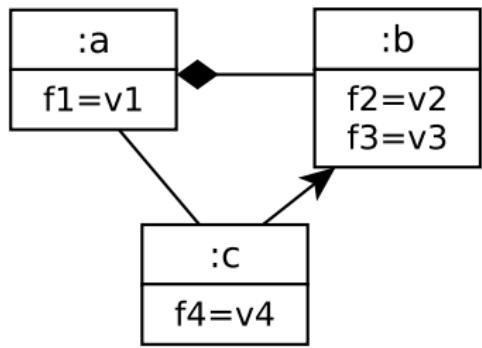
$$\begin{aligned}
 a &= (5, 4, 0, 2, 4, 3, 6, 1) \\
 b &= (6, 5, 3, 3, 4, 7, 0, 1) \\
 d(a, b) &= \frac{1+1+1+1+0+1+1+0}{8} \\
 &= \frac{6}{8}
 \end{aligned}$$

Centralité

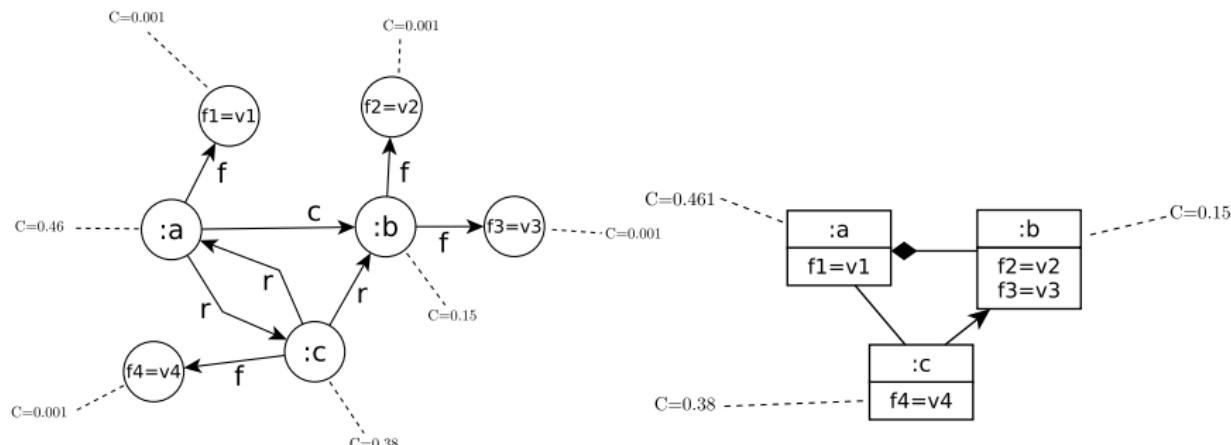
Notre centralité

- Principe de centralité.
- Formule inspirée de page rank de google (principe).
- Adaptée aux modèles.
- Pondération des types, attributs et liens.

Centralité



Centralité

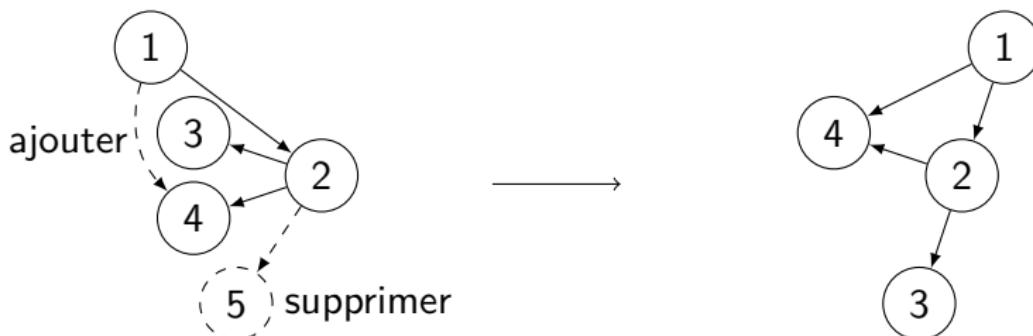


Distance à base de centralité

Calcul

Les deux vecteurs sont comparés par des normes mathématiques (Norme Manhattan, Euclide).

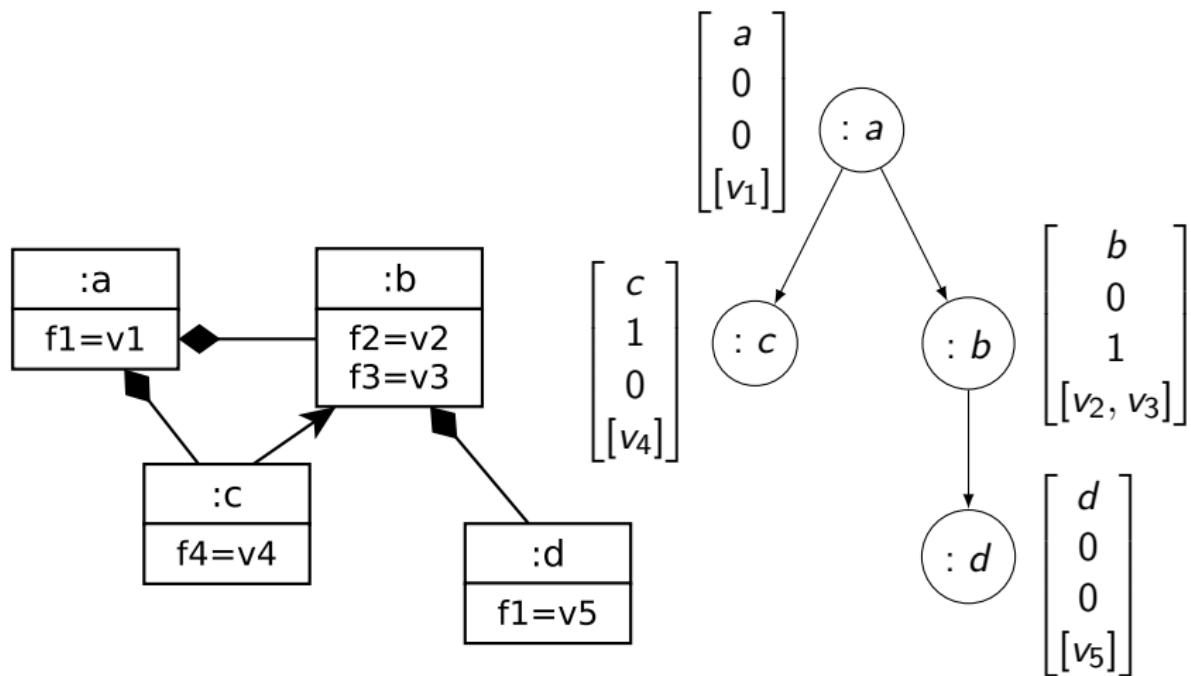
Distance d'édition de modèles



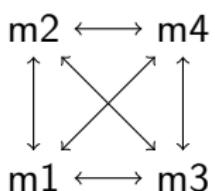
Basée sur

Algorithme de Zhang et Shasha, 89.

Comparaison entre les distances



Diversité d'un ensemble: Matrice de distance et Clustering



| | m1 | m2 | m3 | m4 |
|----|----|----|----|----|
| m1 | 0 | 1 | 5 | 6 |
| m2 | 1 | 0 | 3 | 4 |
| m3 | 5 | 3 | 0 | 10 |
| m4 | 6 | 4 | 10 | 0 |

Objectif

Quels sont les modèles les plus représentatifs ? Quelle est la répartition dans l'espace ?

Modèles les plus représentatifs

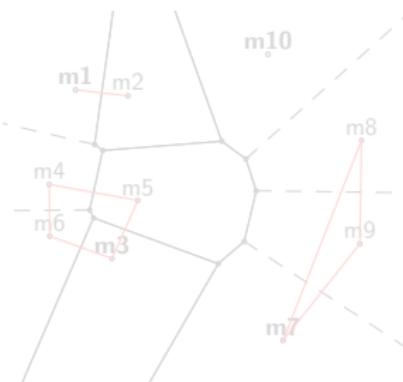
Clustering de la matrice.

| cluster # | Modèles contenus par le cluster |
|-----------|---------------------------------|
| 1 | m7, m8, m9 |
| 2 | m1, m2 |
| 3 | m10 |
| 4 | m3, m4, m5, m6 |

Répartition dans l'espace: Voronoi

Diagramme de Voronoi.

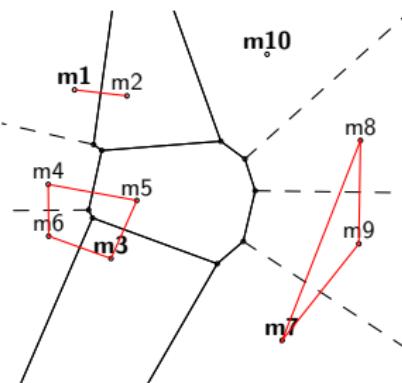
| | m1 | m2 | m3 | m4 | m5 | m6 | m7 | m8 | m9 | m10 |
|-----|----|----|----|----|----|----|----|----|----|-----|
| m1 | 0 | 12 | 27 | 27 | 27 | 26 | 46 | 44 | 45 | 39 |
| m2 | 12 | 0 | 27 | 26 | 27 | 27 | 45 | 45 | 43 | 40 |
| m3 | 27 | 27 | 0 | 18 | 17 | 16 | 46 | 45 | 46 | 39 |
| m4 | 27 | 26 | 18 | 0 | 18 | 18 | 45 | 44 | 45 | 40 |
| m5 | 27 | 27 | 17 | 18 | 0 | 18 | 45 | 43 | 44 | 38 |
| m6 | 26 | 27 | 16 | 18 | 18 | 0 | 45 | 44 | 46 | 40 |
| m7 | 46 | 45 | 46 | 45 | 45 | 45 | 0 | 36 | 36 | 41 |
| m8 | 44 | 45 | 45 | 44 | 43 | 44 | 36 | 0 | 34 | 37 |
| m9 | 45 | 43 | 46 | 45 | 44 | 46 | 36 | 34 | 0 | 39 |
| m10 | 39 | 40 | 39 | 40 | 38 | 40 | 41 | 37 | 39 | 0 |



Répartition dans l'espace: Voronoi

Diagramme de Voronoi.

| | m1 | m2 | m3 | m4 | m5 | m6 | m7 | m8 | m9 | m10 |
|-----|----|----|----|----|----|----|----|----|----|-----|
| m1 | 0 | 12 | 27 | 27 | 27 | 26 | 46 | 44 | 45 | 39 |
| m2 | 12 | 0 | 27 | 26 | 27 | 27 | 45 | 45 | 43 | 40 |
| m3 | 27 | 27 | 0 | 18 | 17 | 16 | 46 | 45 | 46 | 39 |
| m4 | 27 | 26 | 18 | 0 | 18 | 18 | 45 | 44 | 45 | 40 |
| m5 | 27 | 27 | 17 | 18 | 0 | 18 | 45 | 43 | 44 | 38 |
| m6 | 26 | 27 | 16 | 18 | 18 | 0 | 45 | 44 | 46 | 40 |
| m7 | 46 | 45 | 46 | 45 | 45 | 45 | 0 | 36 | 36 | 41 |
| m8 | 44 | 45 | 45 | 44 | 43 | 44 | 36 | 0 | 34 | 37 |
| m9 | 45 | 43 | 46 | 45 | 44 | 46 | 36 | 34 | 0 | 39 |
| m10 | 39 | 40 | 39 | 40 | 38 | 40 | 41 | 37 | 39 | 0 |



Augmenter la diversité: Algorithmique génétique

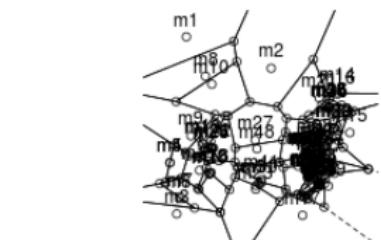
Augmenter la diversité

- Modélisation comme problème d'optimisation en algorithmique génétique.

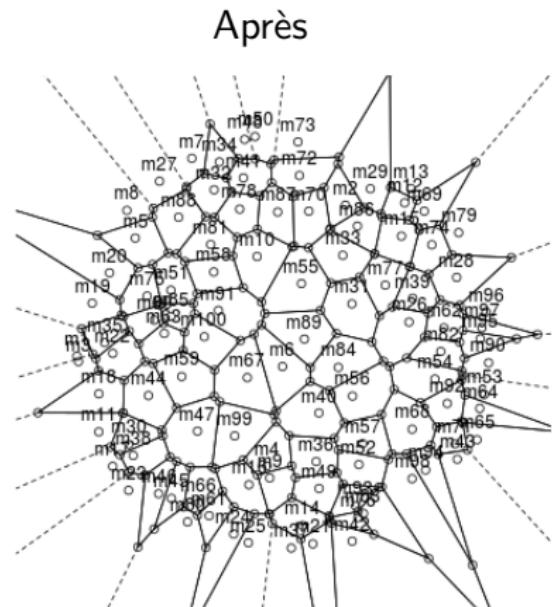
Algorithmique génétique

- Ensemble de modèles = population qui va évoluer pour optimiser un objectif.
- L'objectif = améliorer la diversité par les distances entre modèles.

Augmenter la diversité: Algorithmique génétique

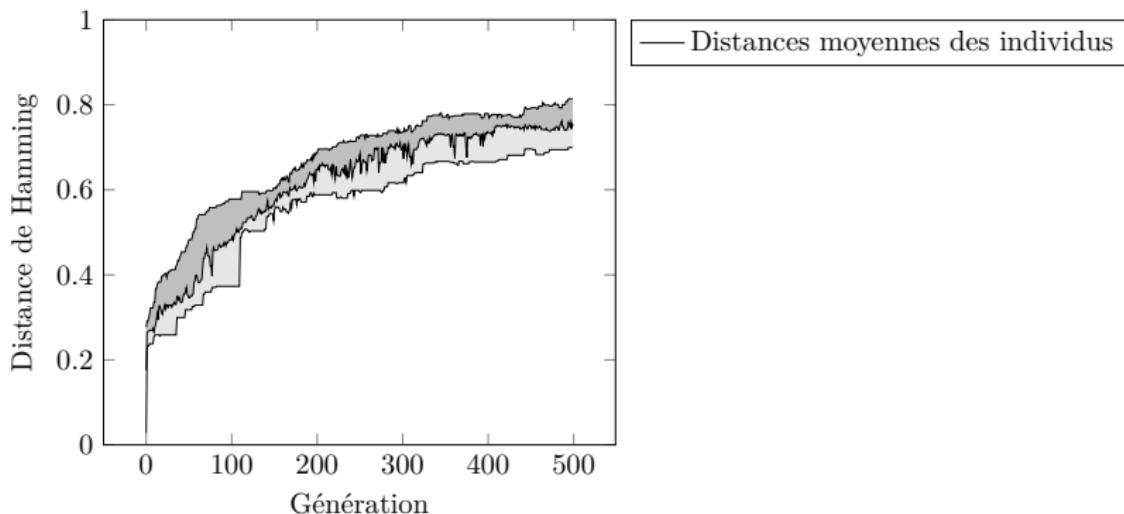


Avant



Après

Augmenter la diversité: Algorithmique génétique

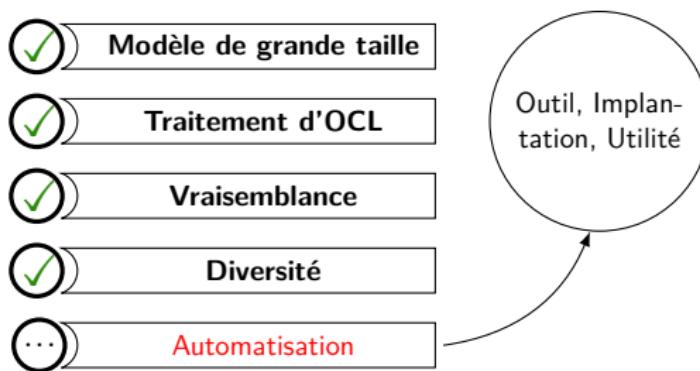


Progression

-  **Modèle de grande taille**
-  **Traitement d'OCL**
-  **Vraisemblance**
-  **Diversité**
-  **Automatisation**

Galinier et al., Meta, 2016.

Progression



Outil GRIMM

Langages et outils

Java/EMF, Xcsp, solveur abscon, R, NSGA.

Utilisation

Interface en ligne de commande, graphique, web.

Utilité

Assistance à la conception de métamodèles, graphes de scaffold pour test.

Outil GRIMM

Langages et outils

Java/EMF, Xcsp, solveur abscon, R, NSGA.

Utilisation

Interface en ligne de commande, graphique, web.

Utilité

Assistance à la conception de métamodèles, graphes de scaffold pour test.

Outil GRIMM

Langages et outils

Java/EMF, Xcsp, solveur abscon, R, NSGA.

Utilisation

Interface en ligne de commande, graphique, web.

Utilité

Assistance à la conception de métamodèles, graphes de scaffold pour test.

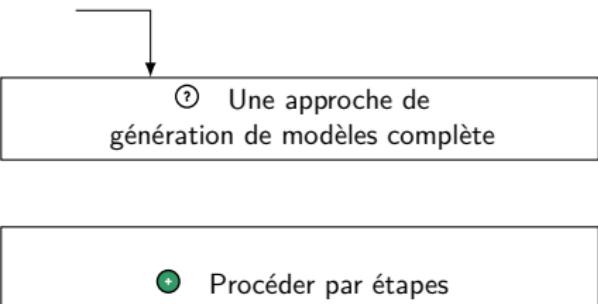
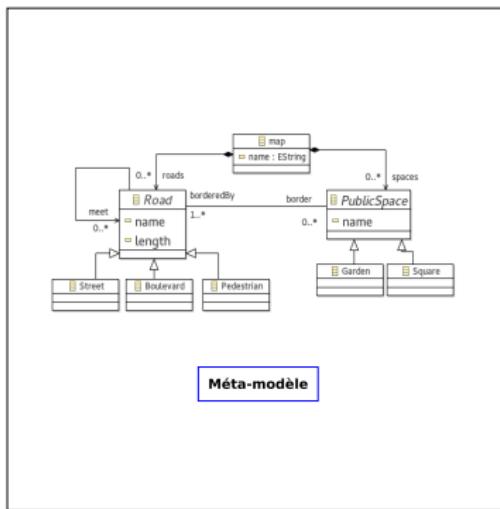
Plan

- ① Ingénierie Dirigée par les Modèles
- ② État de l'art

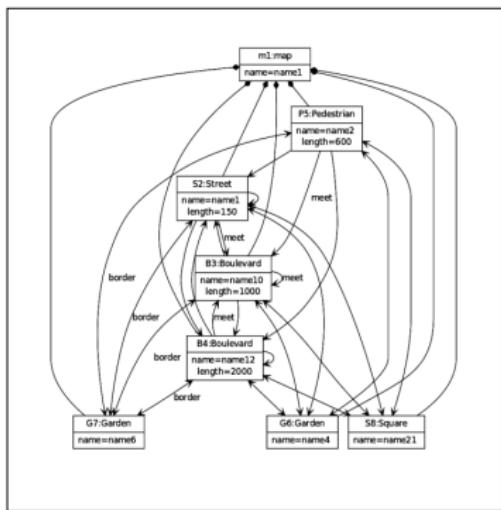
Contributions

- ③ Modèle de grande taille + OCL
- ④ Solutions vraisemblables
- ⑤ Solutions diverses
- ⑥ Conclusion & Perspectives

Conclusion



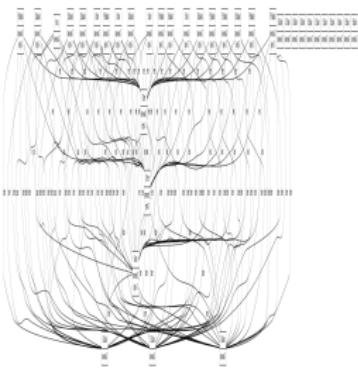
Conclusion



① Générer des instances de métamodèles correctes

② Formaliser les éléments du métamodèle en CSP

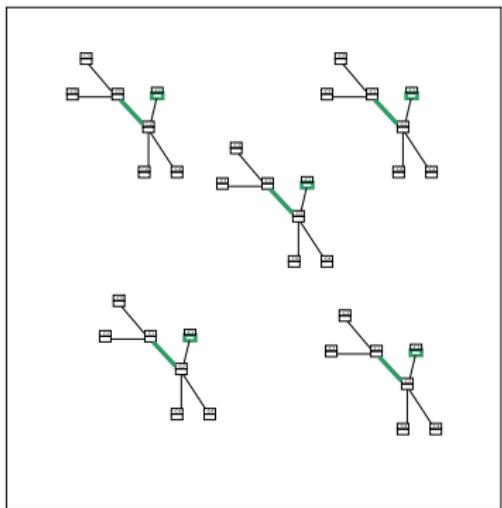
Conclusion



① Passer à l'échelle pour
de plus grandes instances

❷ Utiliser les bonnes pra-
tiques de formalisation en CSP

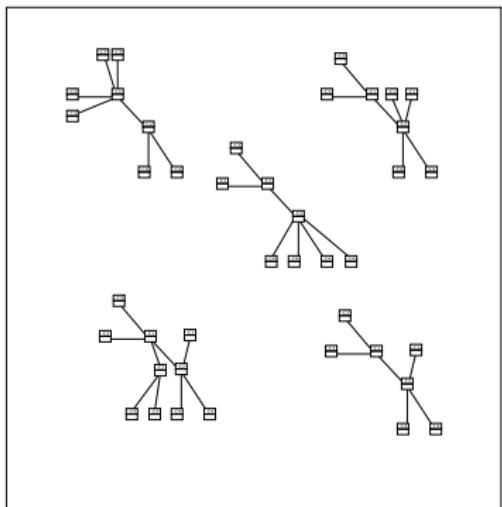
Conclusion



② Rendre les instances con-
formes et valides à la fois

③ Formaliser OCL en CSP
et utiliser les bonnes pratiques

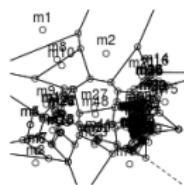
Conclusion



① Rendre les instances plus réalistes, vraisemblables

⊕ Utiliser des métriques et des lois de probabilités

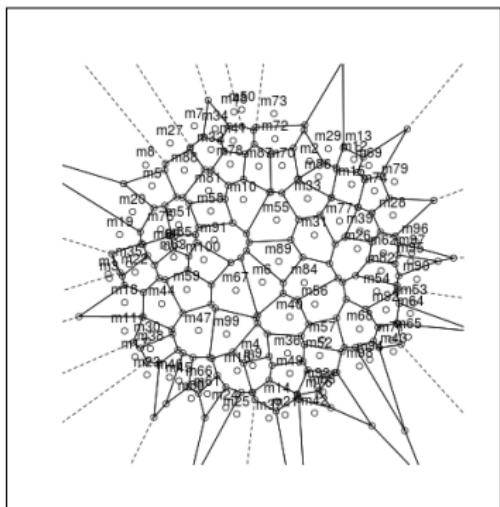
Conclusion



⑦ Comparer des modèles pour quantifier la diversité

- Développer des distances entre modèles et utiliser le clustering

Conclusion



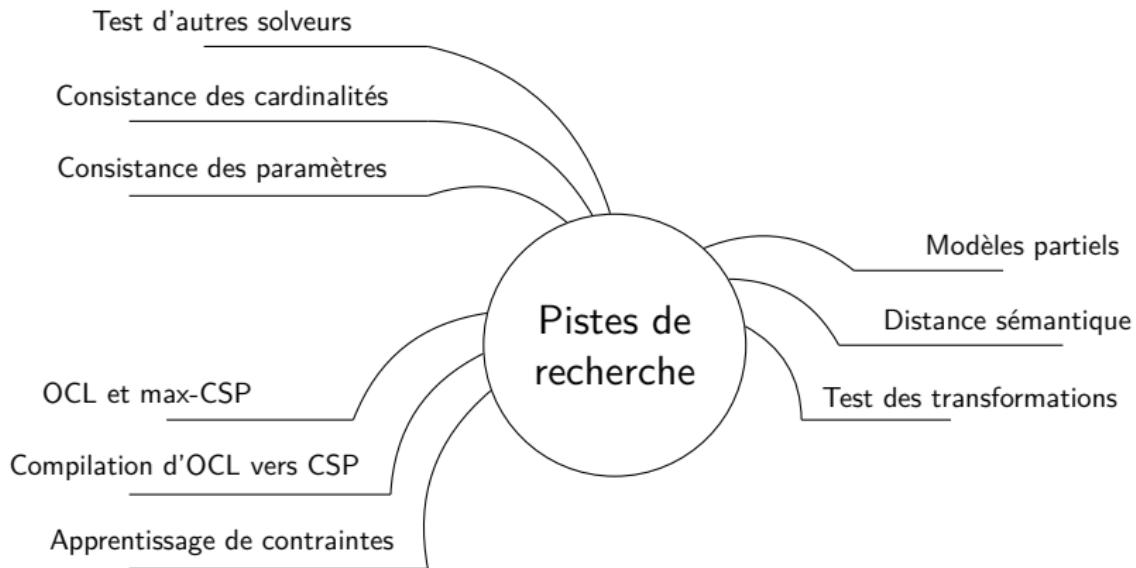
② Améliorer la diversité d'un ensemble de modèles

Modéliser le problème en algorithmique génétique

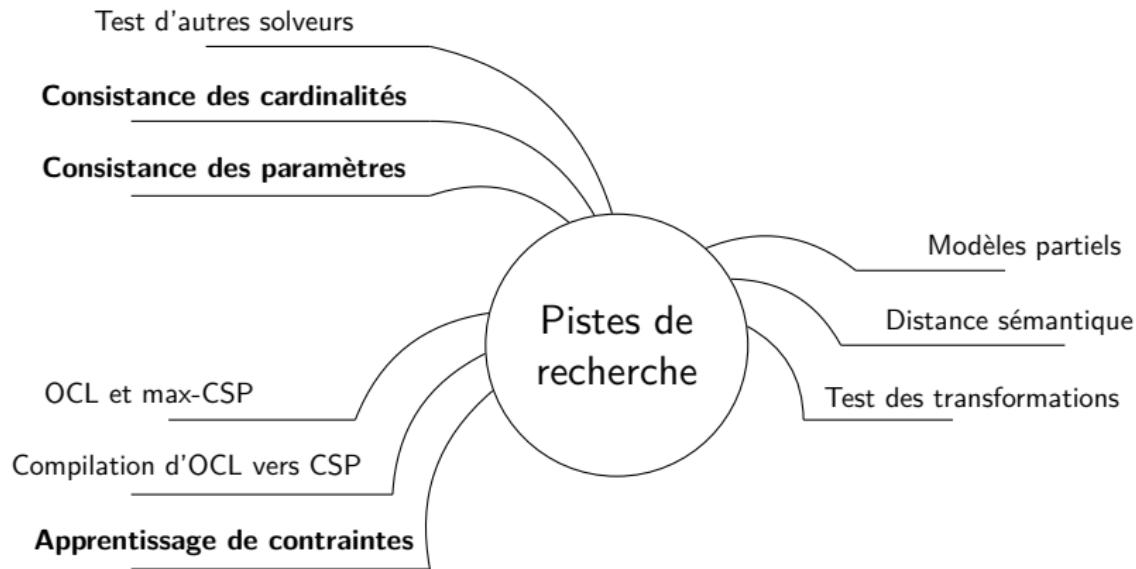
Perspectives



Perspectives



Perspectives



Perspectives

Test d'autres solveurs

Consistance des cardinalités

Consistan

Consistance des cardinalités et des paramètres utilisateur

- La résolution échoue à cause des cardinalités ou des paramètres.
- Vérifier la consistance très tôt avant l'appel au solveur.
- Aider l'utilisateur à corriger ses choix.

OCL et m

Compilation d'OCL vers CSP

Apprentissage de contraintes

les partiels

sémantique

formations

Perspectives

Test d'autres solveurs

Consistance des cardinalités

Consistan

Apprentissage de contraintes

- Un domaine des CSP (équipe Coconut).
- Apprendre par des exemples: Cardinalités, OCL, métriques.

OCL et m

les partiels

sémantique

formations

Compilation d'OCL vers CSP

Apprentissage de contraintes



Tanmirt

+οΙΣΣΟ+