

Nouvelles technologies pour automatiser le développement

Déductions

J.M. Vanel - 2009-05

Appliquer l'intelligence artificielle au génie logiciel

Modélisation, moteurs de règles
multi-modèles, multi-plateformes

Ergonomie et logique

Vision

- Appliquer l'Intelligence Artificielle au développement logiciel
- « Model once, run everywhere »
« logiciel durable »
- Réduire la fracture des plateformes et des langages
 - Les infrastructures ne sont que des commodités, qui doivent être interchangeables

Le besoin des clients

- Vous développez des logiciels spécifiques,
 - Essentiels pour le service offert
- Besoin de réactivité
- Impossible de sous-traiter
- Ne pas être lié à une architecture technique
- Maîtrise et traçabilité dans la chaîne de production
- amical pour l'utilisateur

Notre réponse aux besoins des clients

- Environnement de développement à base de règles (EulerGUI)
- 3 moteurs d'inférence (Java-Drools, Prolog, Python-CWM)
- lit des modèles métiers dans plusieurs formats (UML, OWL, RDF(S), SQL, eCore)
- Exemple d'utilisation: générateur de formulaires
 - Règles logiques inhérentes ==> ergonomie supérieure

Moteurs de règles: avantages

- Une règle isolée a un sens
- Le moteur enchaîne les règles, pas le programmeur
- + efficace que la programmation classique
- Capture bien l'expertise métier
- Traduisible depuis et vers le langage naturel
- Les « systèmes experts » ont fait leurs preuves

Règles: exemples

S'il y a une commande, il y a une demande de paiement associée :

```
{ ?C a Order } =>
```

```
{ ?C hasPaymentRequest ?PR } .
```

Si le porteur a eu un incident de paiement, alors la commande est annulée:

```
{ ?C hasHolder ?H .
```

```
  ?H hasPaymentIncident ?PI } =>
```

```
{ ?B hasStatus cancelled } .
```

Modélisation facilitée – UML textuel

```
Person<|---Customer---Buying<|---ServiceBuying  
                                <|---MerchandiseBuying
```

```
HelpRequest-----Buying  
date: Date  
explanation: String
```

Modélisation facilitée – Anglais contrôlé

Every card is a means of a payment .

Le modèle formel est récupéré à partir de la phrase:

```
Card ---|> MeansOfPayment
```

Générateur de formulaire

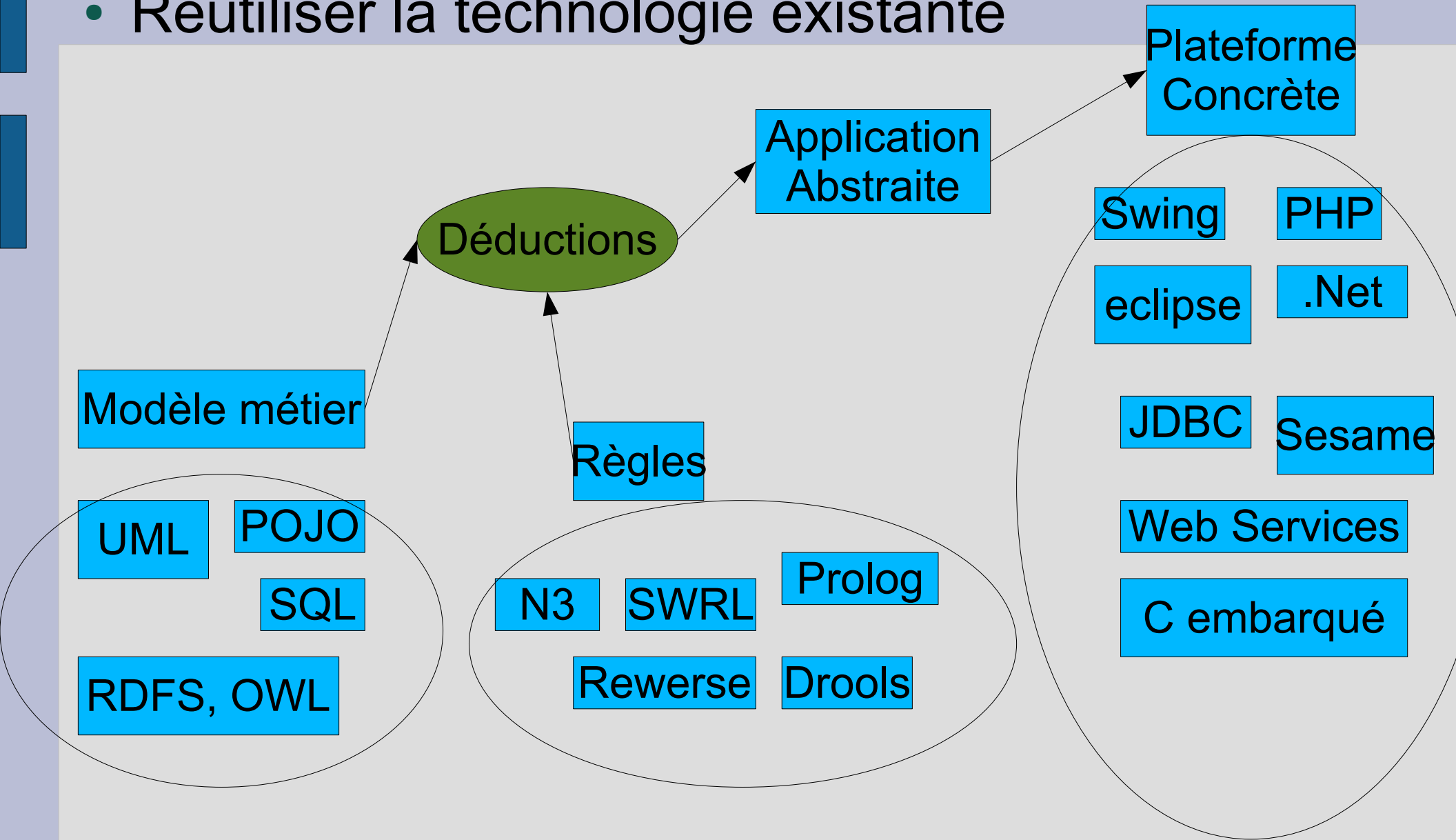
- Les techniques courantes des générateurs d'application sont limitées:

- | Générateur X | Déductions |
|--|---|
| | Permet de générer 100% d'une application à partir du modèle et des règles |
| Couteux en technicité | |
| Formats spécifiques ou propriétaires en entrée | S'appuie sur une modélisation riche, mais standard : OWL du W3C |
| plateforme spécifique en sortie | Conception multi-plateforme |
| Coûts de licence | Open Source |

- OWL est mieux adapté qu'UML pour modéliser les concepts métier
- OWL est basé sur la logique de description

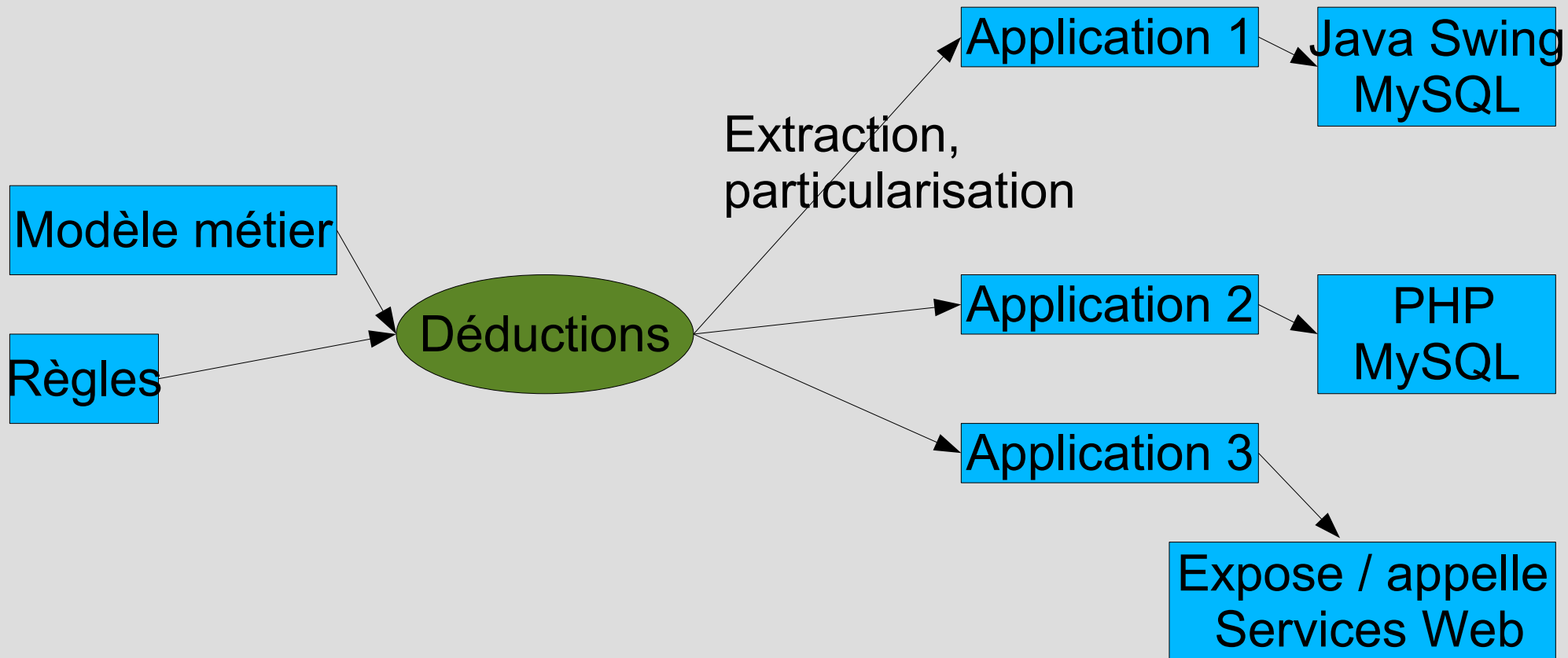
Le flux d'information (vu par le concepteur)

- Réutiliser la technologie existante



Cohérence et réutilisation

- Réutiliser les modèles existants



Le Web Sémantique

- Fondement solide pour les modèles métier
 - Standards d'échange: RDF, OWL, N3
 - Milliers de modèles disponibles sur plusieurs moteurs de recherche
 - Au contraire UML est complexe, peu interopérable, sans modèles disponibles
- Le WS est pensé pour exposer des données riches sur le Web; nous le réutilisons comme socle pour l'ingénierie du logiciel