

Optimisation de la compilation de règles métier probabilistes à l'aide de PRM

Gaspard Ducamp^{1,2}, Philippe Bonnard², Christian de Sainte Marie²,
Christophe Gonzales¹, Pierre-Henri Wuillemin¹

¹LIP6 (UMR 7606), Sorbonne Université, 4 place Jussieu, 75005 Paris, France
{prenom.nom}@lip6.fr

²IBM France Lab, 9 rue de Verdun, 94250 Gentilly, France
philippe.bonnard@fr.ibm.com, csma@fr.ibm.com
gaspard.ducamp@ibm.com

Cadre de la thèse

IBM France Lab (anciennement ILOG)

Laboratoire d'Informatique de Paris VI
Sorbonne Université

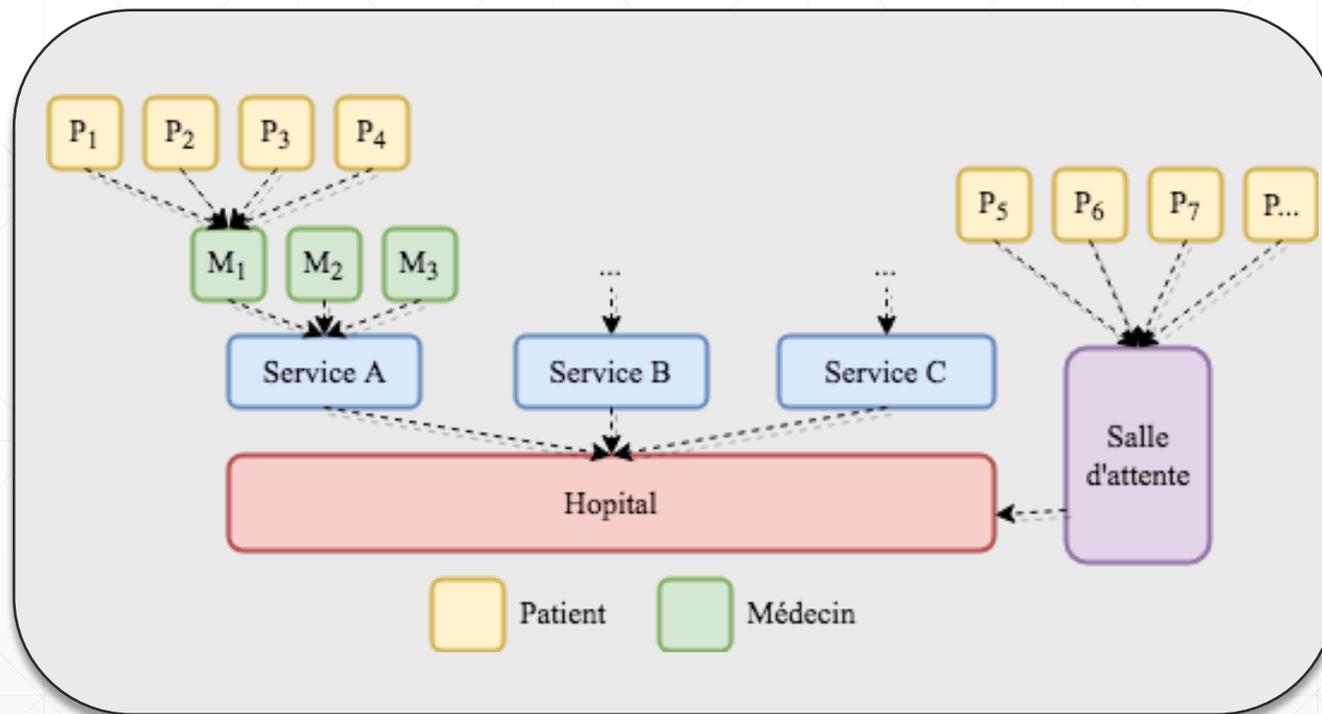
- Etudes PPR (IBM)
- Thèse URBS (Hamza Agli, 2017)

Plan

1. Contexte
 - Règles métiers et BRMS
 - Probabilistic Relational Model
 - Enjeux et objectifs
2. Etat actuel
3. Nos propositions
4. Ouvertures

Exemple d'utilisation

- Optimiser la gestion des patients arrivant dans un établissement hospitalier
- Utiliser des modèles probabilistes pour mieux cibler les besoins les patients



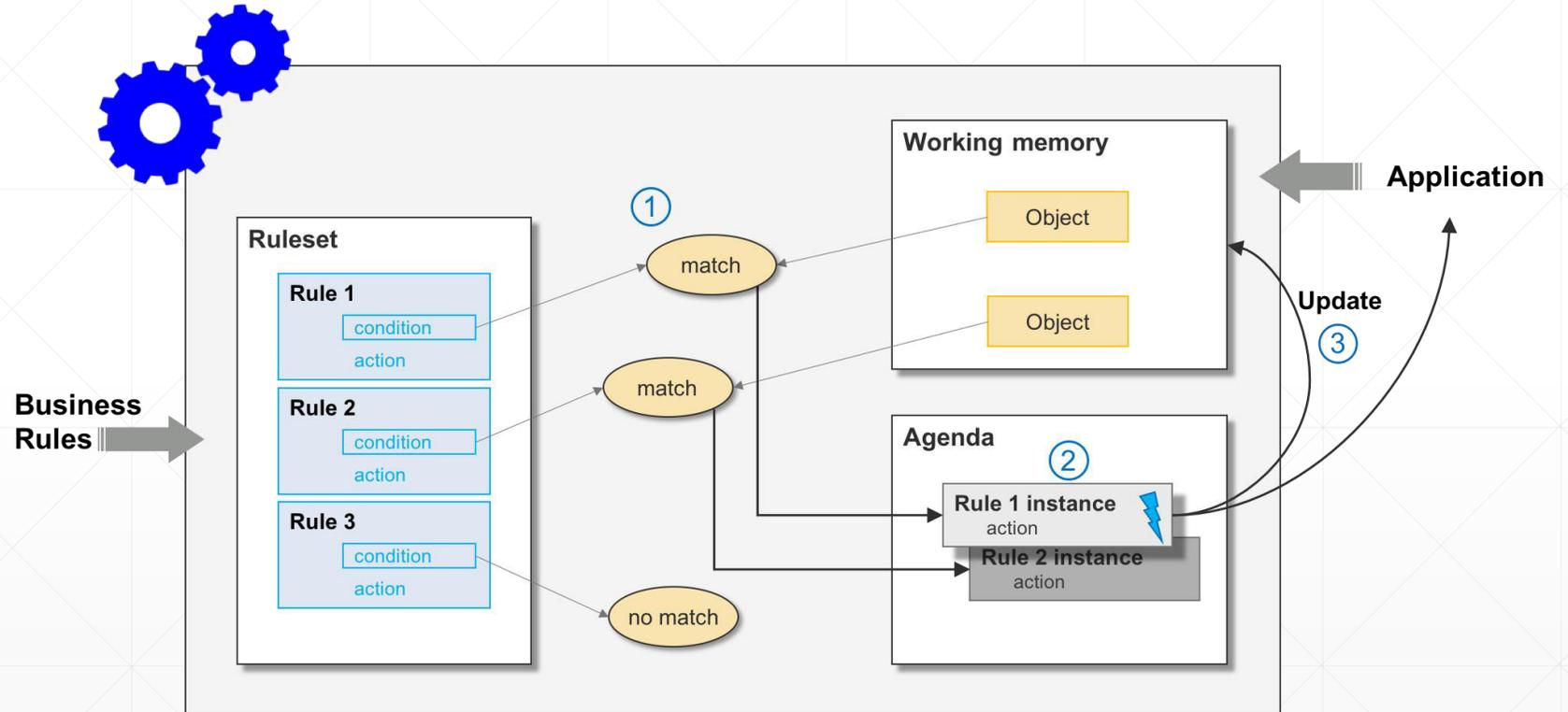
Règles métiers

Si un patient à besoin d'un médecin rapidement et qu'un médecin est disponible
Alors on lui transfère le patient

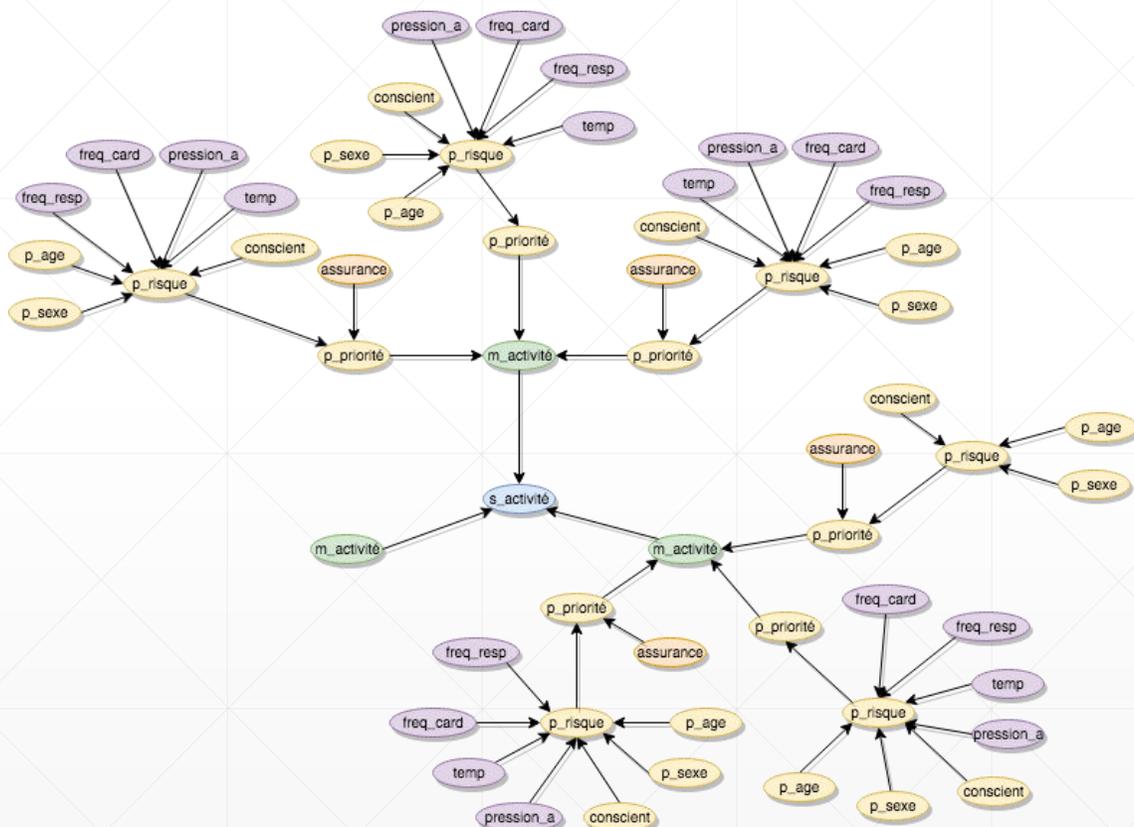
- Permet de traduire les politiques métiers des entreprises (banque, assurance,...)
- Utilisation d'une syntaxe proche du langage métier

Systemes expert à base de règles

- Un système expert à base de règle c'est:
 - Un ensemble de règles (Ruleset)
 - Un ensemble d'objets (Working memory)
 - Un moteur d'inférence par chaînage avant



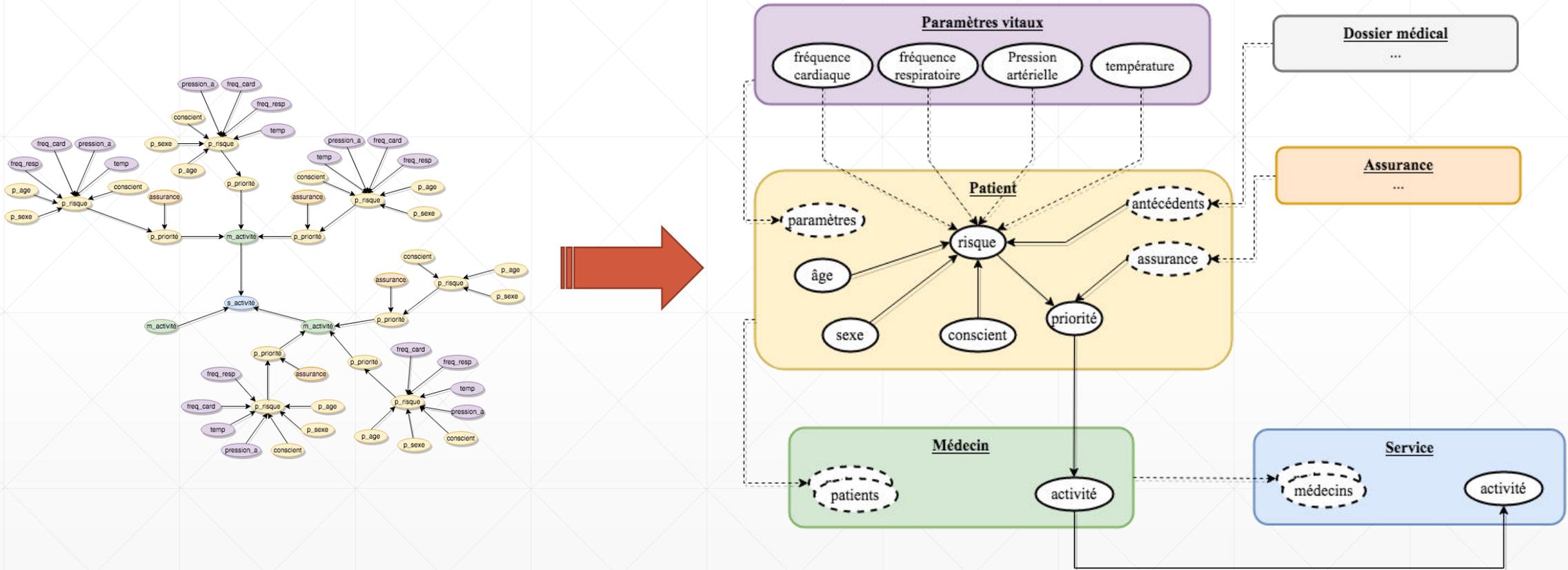
Probabilistic Relational Model



La modélisation et la manipulation de grands BN est fastidieuse:

- Pourquoi ne pas utiliser la redondance structurelles des éléments pour les faciliter?
- Pour accélérer les calculs ?

Probabilistic Relational Model



Enjeux et objectifs

Besoin industriel d'IBM :

- Enrichir ODM en proposant des règles métier probabilistes

Utilisation d'aGrUM :

- Enrichir la capacité de modélisation
- Travailler sur des algorithmes spécifiques

Pour ce faire :

- Couplage BRMS/PGM

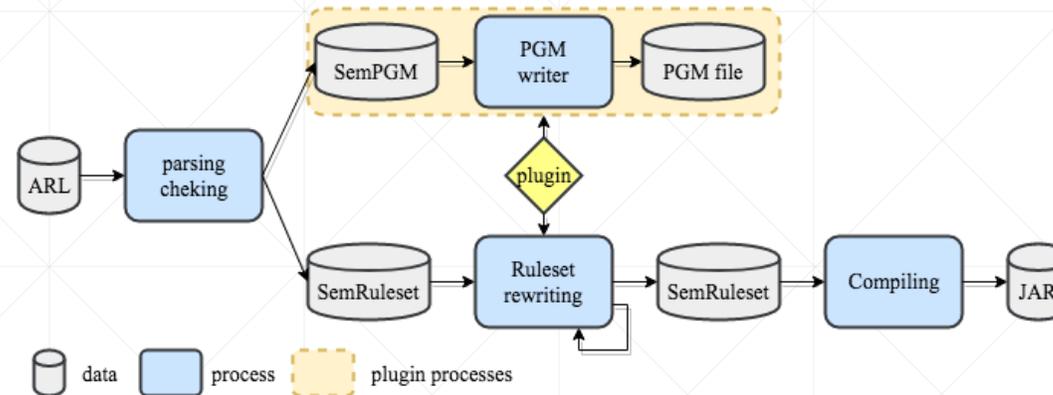
Une première approche

- Permet de faire des requêtes probabilistes sur des prédicats particuliers

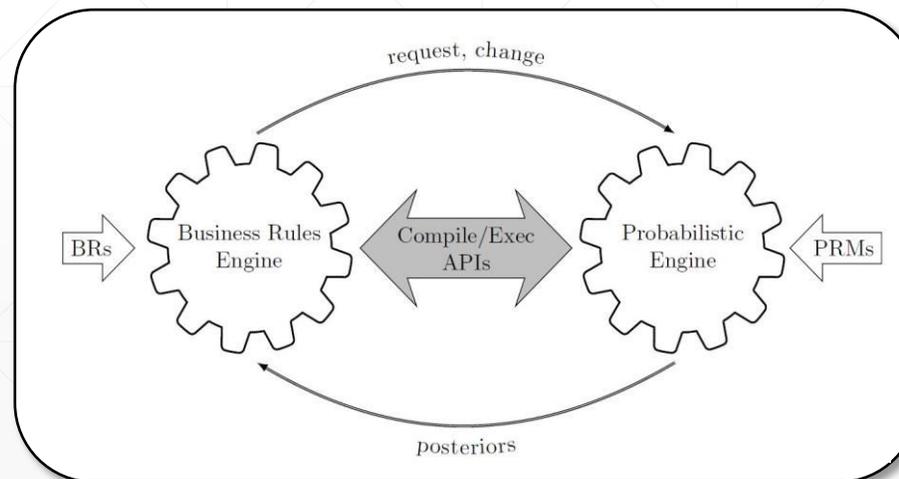
```
Règle r1{  
  si {  
    p : Patient(probabilite(p.priorité<=moyen)>.9)  
    m : Medecin(size(m.patients)==0);  
  } alors {  
    m.patients.add(p)  
  }  
}
```

Une première approche

- Lors de la compilation d'ODM, on fait le lien entre le BRMS et le moteur probabiliste choisi



- Lors de l'exécution, on a un couplage entre moteur de règles et moteur probabiliste



Limitations

- La définition de règles probabilistes est trop compliquée pour un utilisateur métier, une expressivité aussi fine demande une connaissance poussée du modèle probabiliste sous-jacent,
- Les derniers travaux se sont centrés sur la création d'un nouvel algorithme d'inférence (IJTI), dont le cadre d'application est trop spécifique,
- IJTI travaille avec des PRM mais retranscrits sur la forme de BN dits « groundés ». On n'exploite pas les redondances structurelles encodées dans les PRM.

Vers une nouvelle définition des règles métiers probabilistes

```

Règle r1{
  si {
    p : Patient(p.priorité<=moyen)
    m : Medecin(size(m.patients)==0);
  } [avec probabilité >.8] alors {
    m.patients.add(p)
  } sinon {
    ...
  }
}

```

Si on pense à 80% qu'un nouveau patient a une priorité au plus modérée et qu'un médecin est libre **alors** on lui assigne le patient.

```

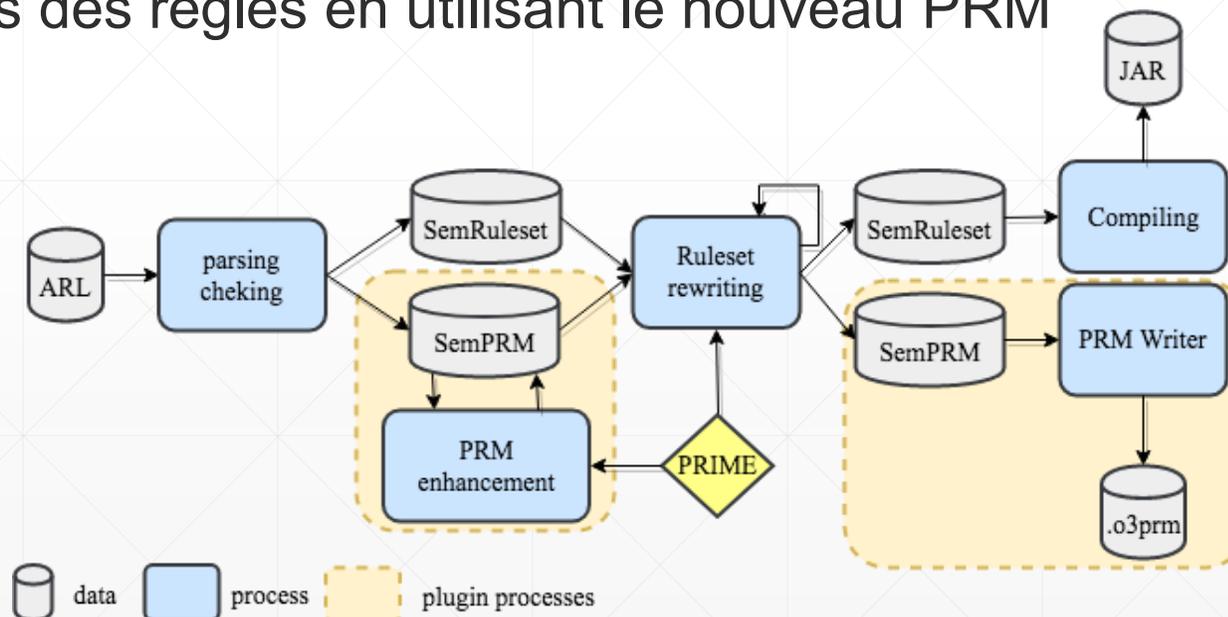
Règle r2{
  si {
    p : Patient(p.priorité==élevé and
    p.conscient==faux)
    m : Medecin(m.activité<=moyen);
  } [avec probabilité >.9] alors {
    m.patients.add(p)
  } sinon {
    ...
  }
}

```

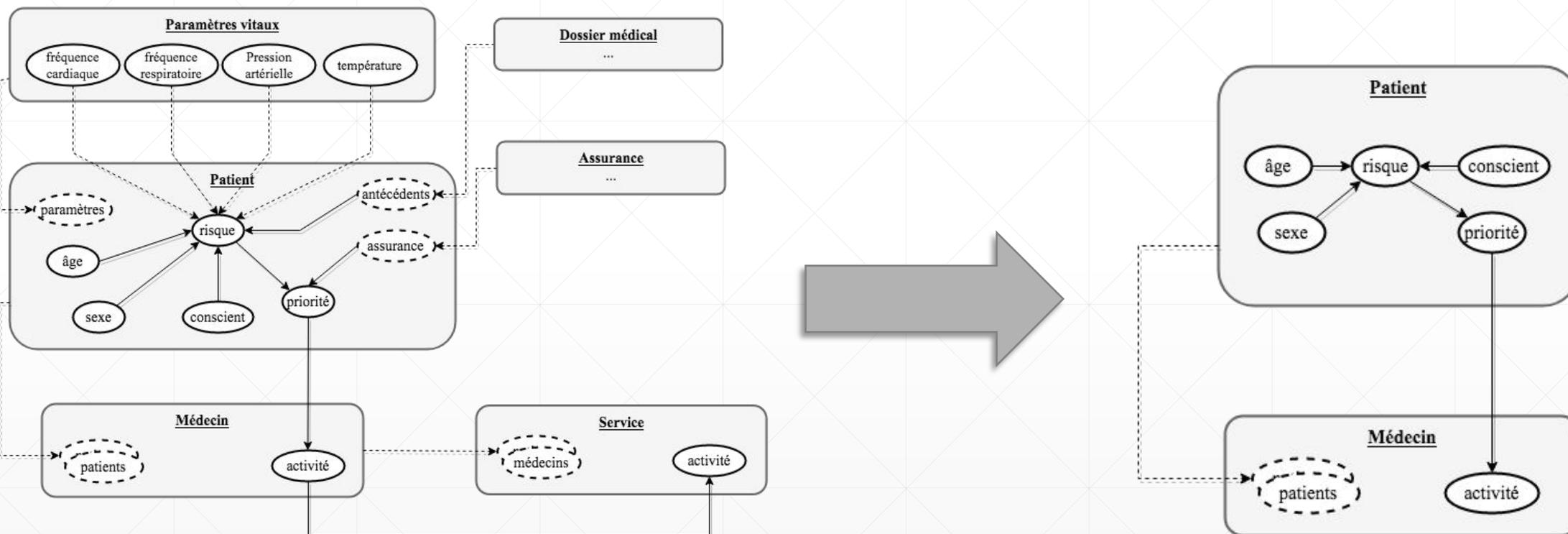
Si on pense à 90% qu'un patient inconscient a une priorité élevée et un médecin ne traite que des patients aux risques faibles ou modérés **alors** on lui assigne le patient.

Nos propositions

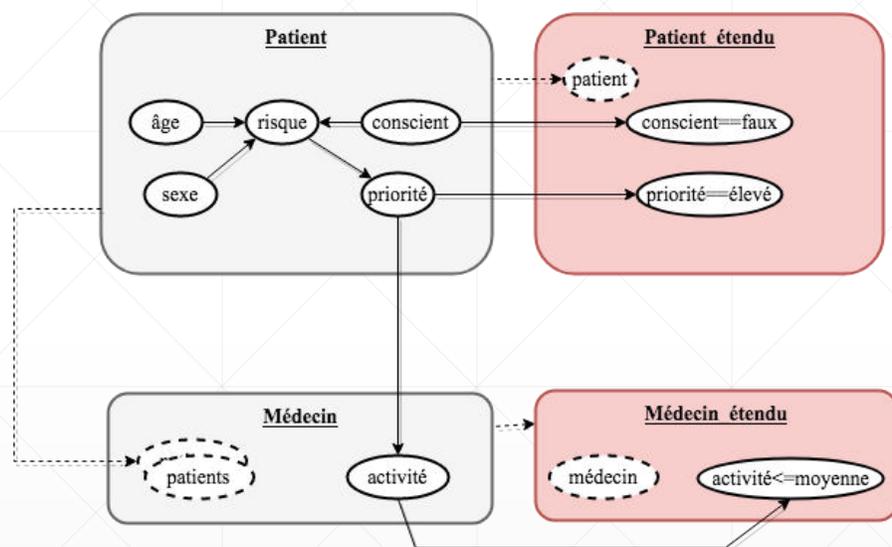
- Utilisation du Ruleset pour compléter le PRM lors de la compilation **avant** la réécriture
- Réécritures des règles en utilisant le nouveau PRM



Augmentation du PRM



Augmentation du PRM



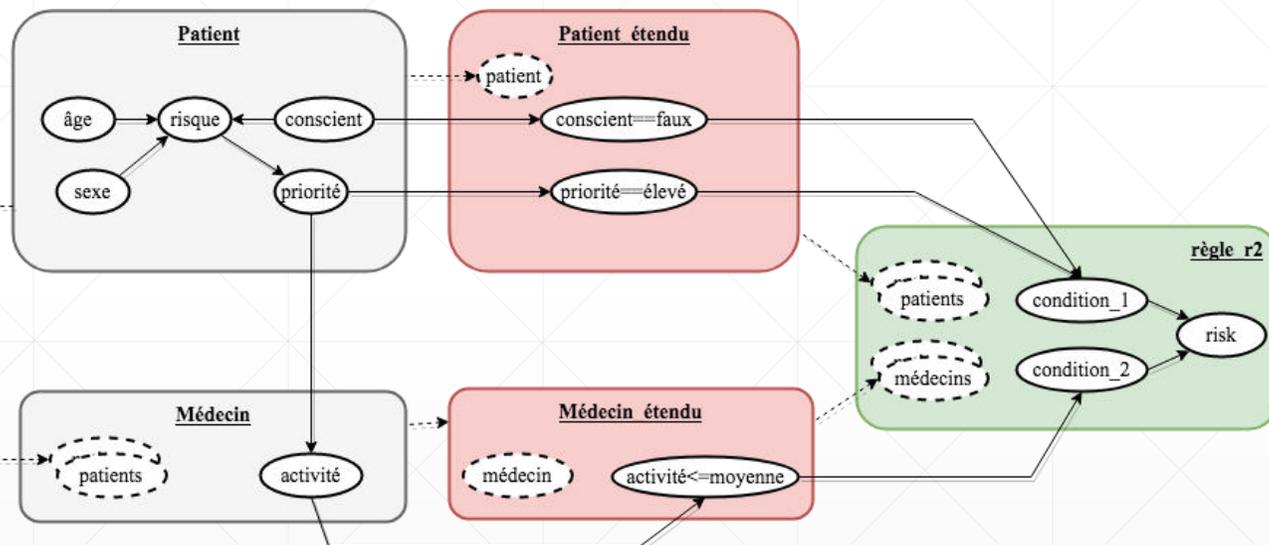
```

Règle r2{
si {
  p : Patient(p.priorité==élevé and p.conscient==faux)
  m : Medecin(m.activité<=moyen);
} [avec probabilité >.9] alors {
  m.patients.add(p)
} sinon {
  ...
}
}

```

- On étudie la liste des prédicats pour augmenter les classes concernées

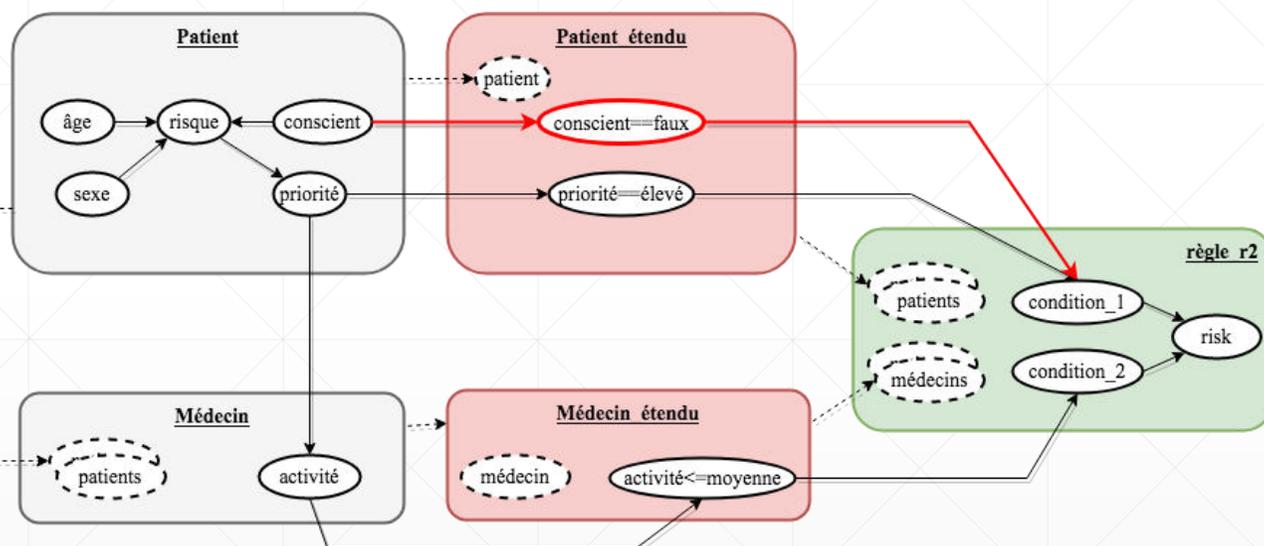
Augmentation du PRM



```

Règle r2{
si {
    p : Patient(p.priorité==élevé and p.conscient==faux) (c1)
    m : Medecin(m.activité<=moyenne); (c2)
} [avec probabilité >.9] alors {
    m.patients.add(p)
} sinon {
    ...
}
}
    
```

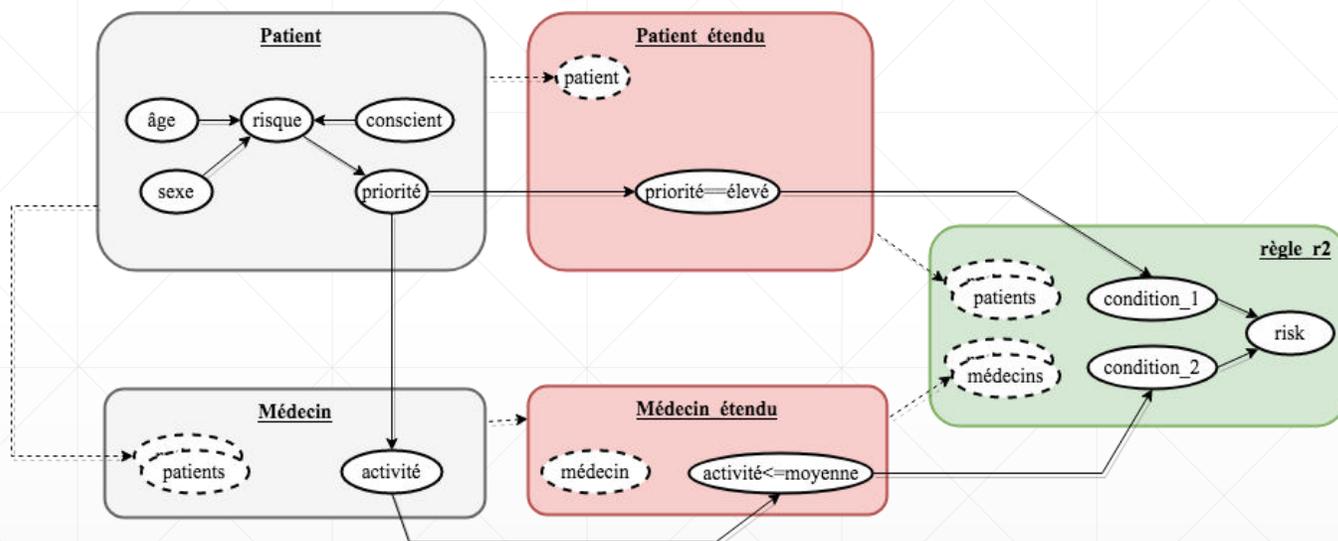
Simplification du modèle



```

Règle r2{
si {
    p : Patient(p.priorité==élevé and p.conscient==faux) (c1)
    m : Medecin(m.activité<=moyen); (c2)
} [avec probabilité >.9] alors {
    m.patients.add(p)
} sinon {
    ...
}
}
    
```

Réécriture de la règle



```

Règle r2{
si {
    r: règle_r2(probabilité(r.risk)>.9);
} alors {
    m.patients.add(p)
} sinon {
    ...
}
}
    
```

Difficultés

- Identifier les différentes constructions possibles et les étudier au cas par cas (explosion combinatoire due aux agrégateurs, par exemple)
- Impact sur le pattern matching opéré par le BRMS
- Travail sur la réduction de la complexité du PRM lors de la compilation (simplifier prédicats déterministes)

Ouvertures

- Étudier la gestion de règles temporelles probabilistes, comme ce qui se fait en traitement des évènements complexes (*CEP*)
- Étudier de nouvelles approches pour la quantification du risque ?
- Travailler sur un nouvel algorithme d'inférence pour aGrUM prenant en compte la nature redondante des modèles utilisés.



Merci de votre attention !