

G.T. Big Data – 4 avril 2024

C. Rey

Activités de recherche




Généralités

- Recherche en IA symbolique
- Représentation des connaissances et raisonnements (KRR)
 - Logiques de description (peu expressives)
 - Programmation logique (prolog, WFS, un peu ASP)
- Nature des recherches
 - 1/3 de théorie
 - 1/3 d'application (aéronautique, e-santé, jeux vidéo, ...)
 - 1/3 d'implémentation

Problématiques et collaborations

- Raisonnements de différence et de matchmaking avec des logiques de description peu expressives


Différence dans ALN et EL



- Ma thèse et travaux ultérieurs
- Thèse A. Mascaro (2019-2023), projet STAM (métrologie) - Collaboration **Perfect Memory** - Financement FEDER

4

Matchmaking par meilleure couverture
Matchmaking multicritère



- Ma thèse et travaux ultérieurs
- Thèse A. Mascaro (2019-2023), projet STAM (métrologie) - Collaboration **Perfect Memory** - Financement FEDER

8


- Méthodologies de programmation déclarative à l'aide de règles et/ou bases de connaissances

Rappels sur l'intégration de données et l'OBDA



12

Intégration de données déclarative par réécriture de requêtes



- Thèse de John Samuel (2011-2014) - Collaboration **Rootsystem** - Financement Région Auvergne et FEDER
- Postdoctorat Karima Ennaoui (2019-2021), Ingénieur d'études Mathieu Faivre (2019-2020) - Collaboration **Dassault Aviation** - Financement DGA

19

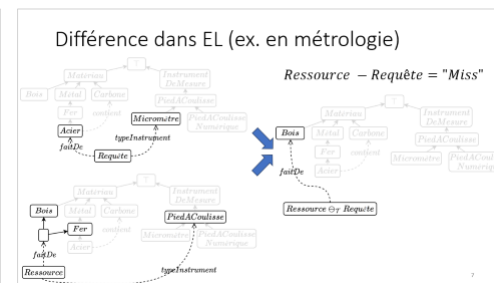
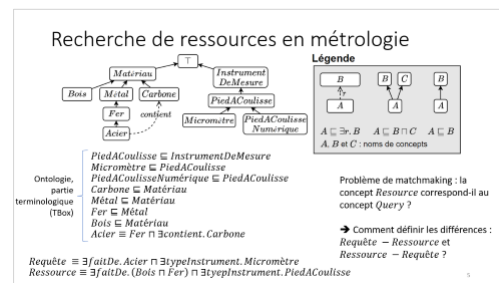
Recommandation et prise de décision déclarative à l'aide d'ontologies et de règles



- Ingénieur d'études Adrien Collange (2022-2024) - Collaboration **Jeolis Solutions** - Financement ANR Plan de relance
- Thèse de Syvain Lapeyrade (2020-...) - Collaboration **Wako Factory** - Financement ANR Contrat doctoral en IA

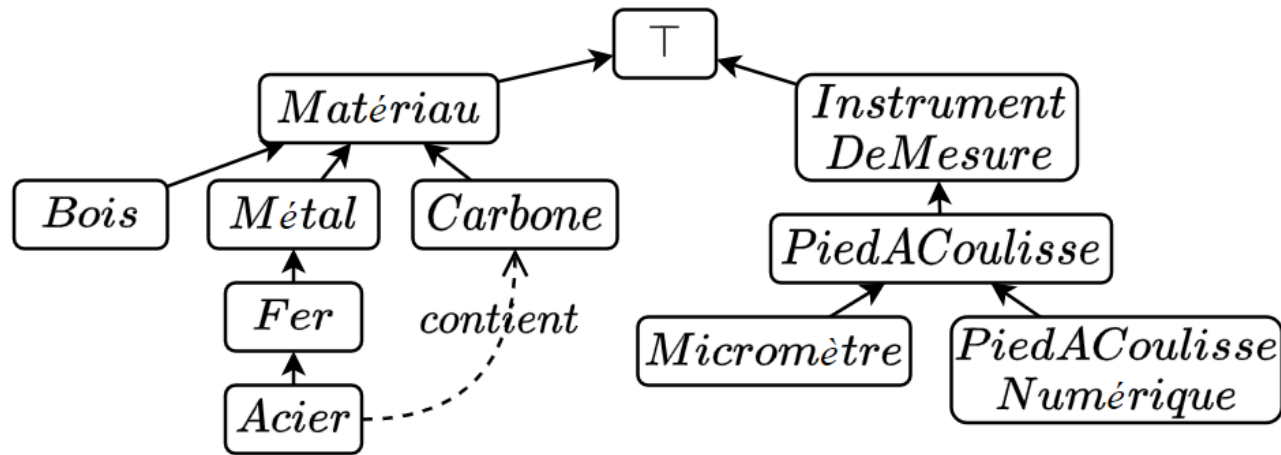
22

Différence dans ALN et EL

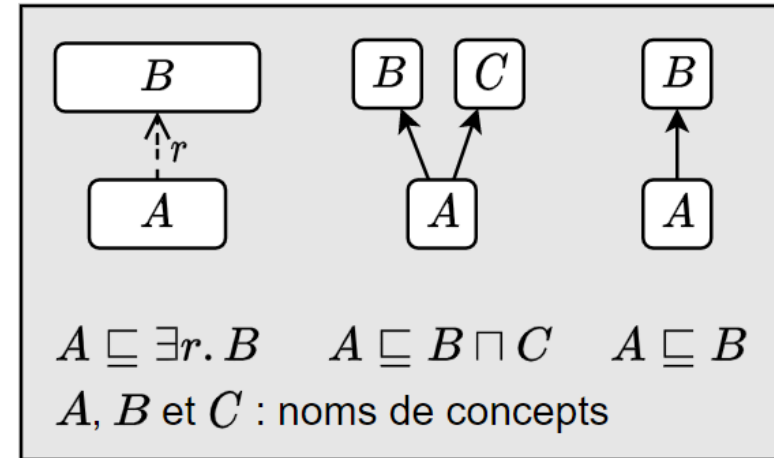


- Ma thèse et travaux ultérieurs
- Thèse A. Mascaro (2019-2023), projet STAM (métrologie) - Collaboration **Perfect Memory** - Financement FEDER

Recherche de ressources en métrologie



Légende



Ontologie,
partie
terminologique
(TBox)

$PiedACoulisse \sqsubseteq InstrumentDeMesure$
 $Micromètre \sqsubseteq PiedACoulisse$
 $PiedACoulisseNumérique \sqsubseteq PiedACoulisse$
 $Carbone \sqsubseteq Matériau$
 $Métal \sqsubseteq Matériau$
 $Fer \sqsubseteq Métal$
 $Bois \sqsubseteq Matériau$
 $Acier \equiv Fer \sqcap \exists contient. Carbone$

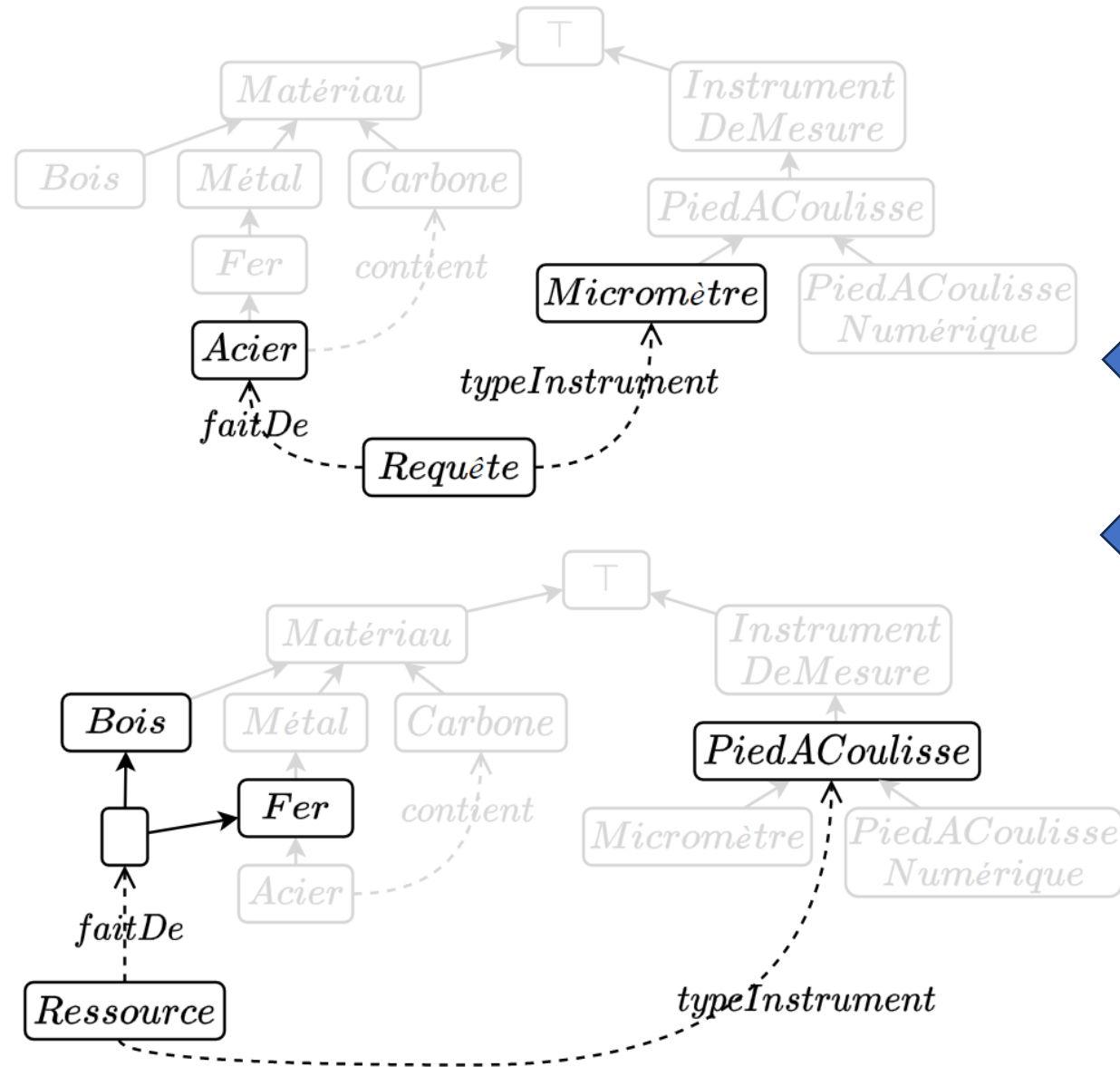
$Requête \equiv \exists faitDe. Acier \sqcap \exists typeInstrument. Micromètre$

$Ressource \equiv \exists faitDe. (Bois \sqcap Fer) \sqcap \exists typeInstrument. PiedACoulisse$

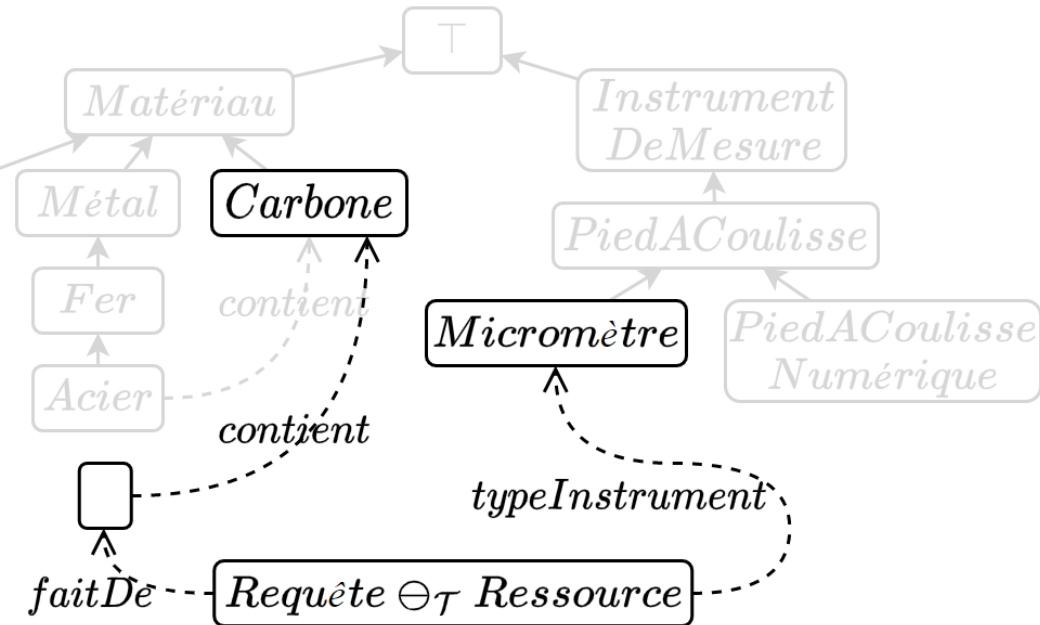
Problème de matchmaking : la
concept *Resource* correspond-il au
concept *Query* ?

➔ Comment définir les différences :
Requête – *Ressource* et
Ressource – *Requête* ?

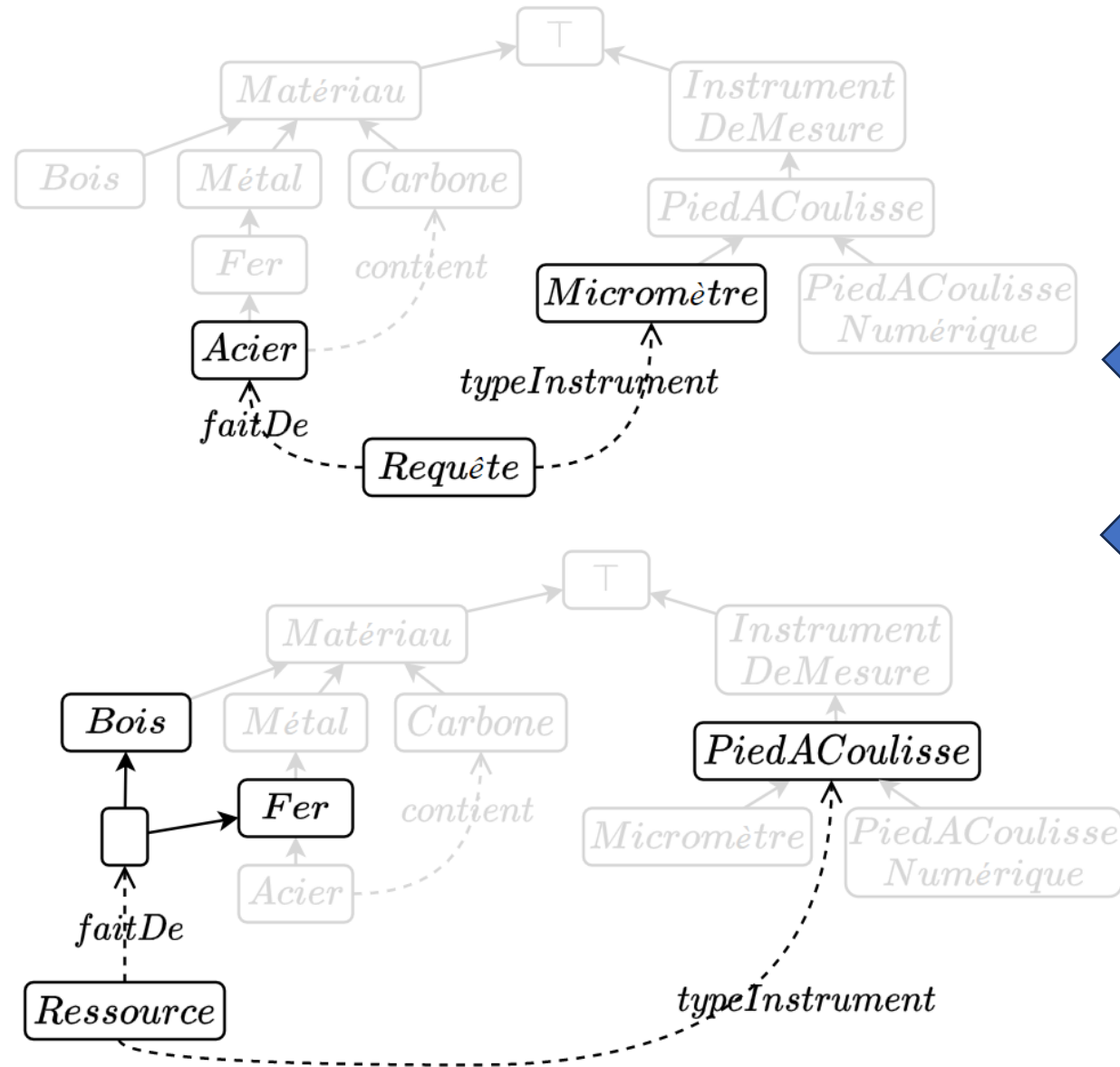
Différence dans EL (ex. en métrologie)



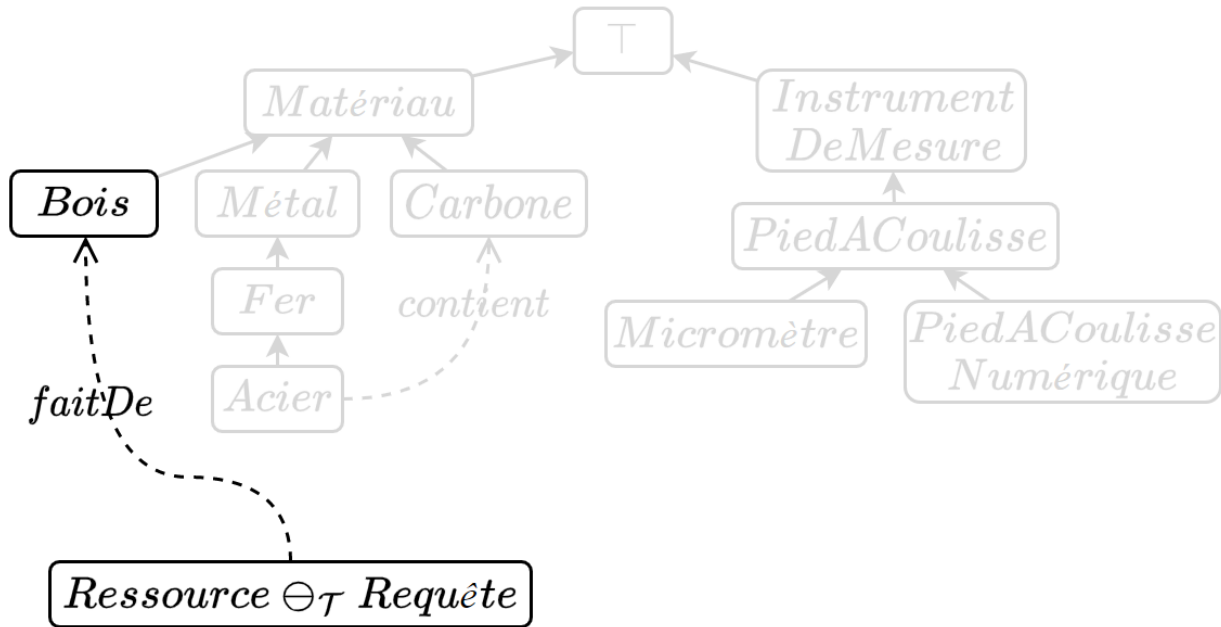
Requête – Ressource = "Rest"



Différence dans EL (ex. en métrologie)

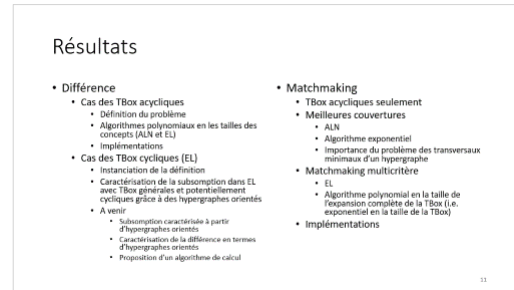
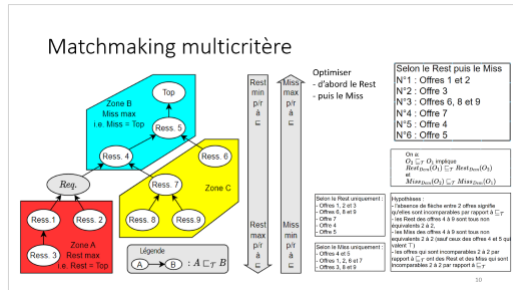
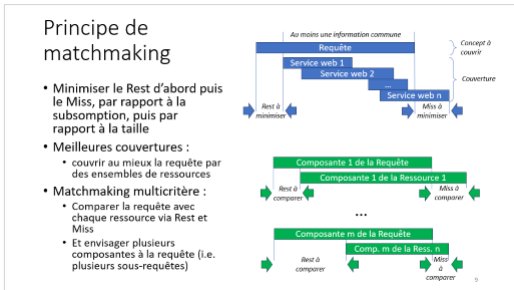


Ressource – Requête = "Miss"



Matchmaking par meilleure couverture

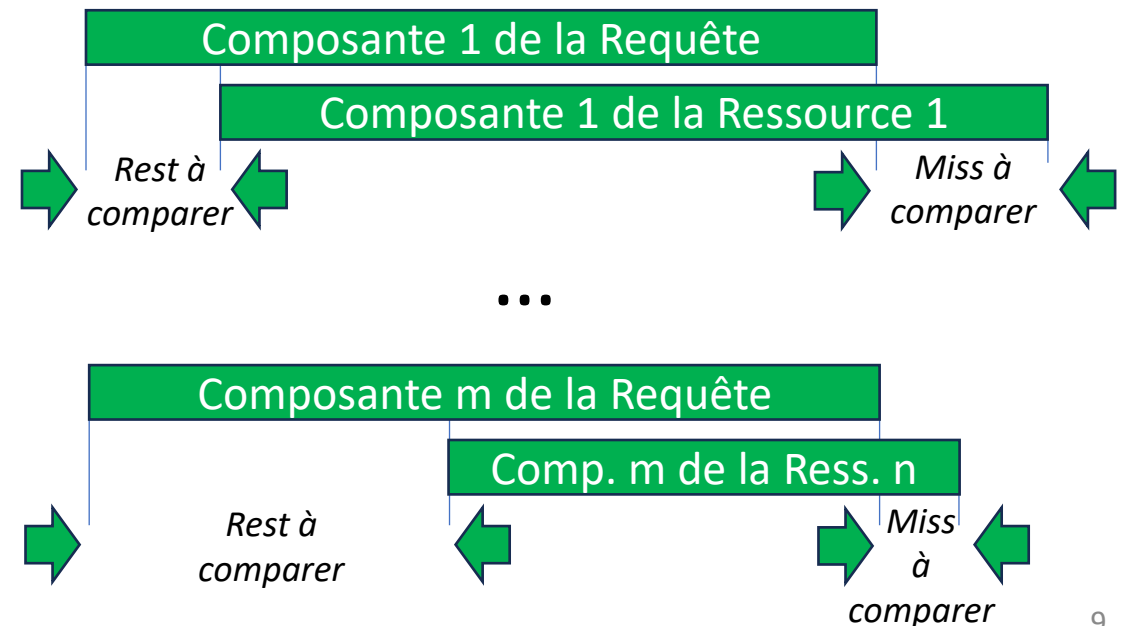
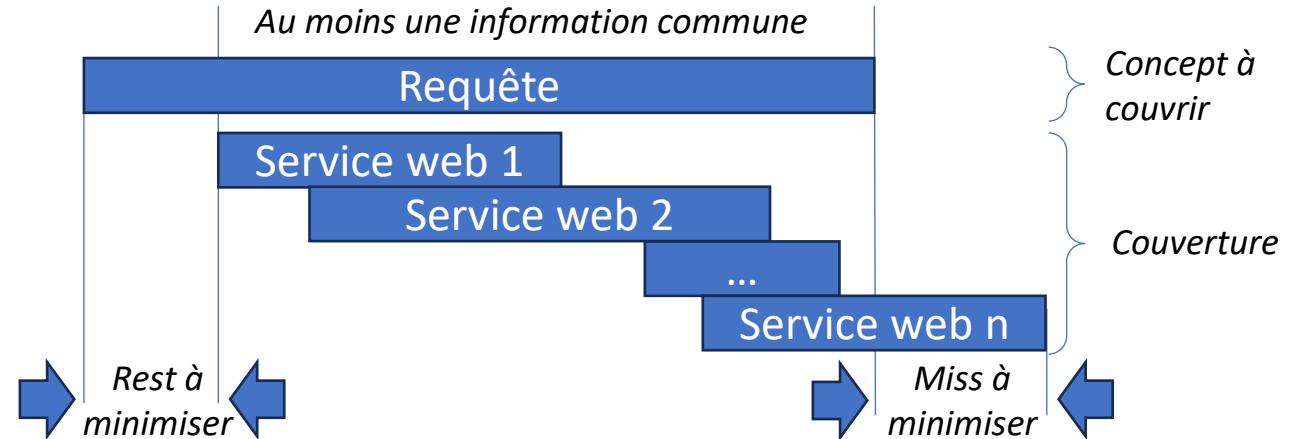
Matchmaking multicritère



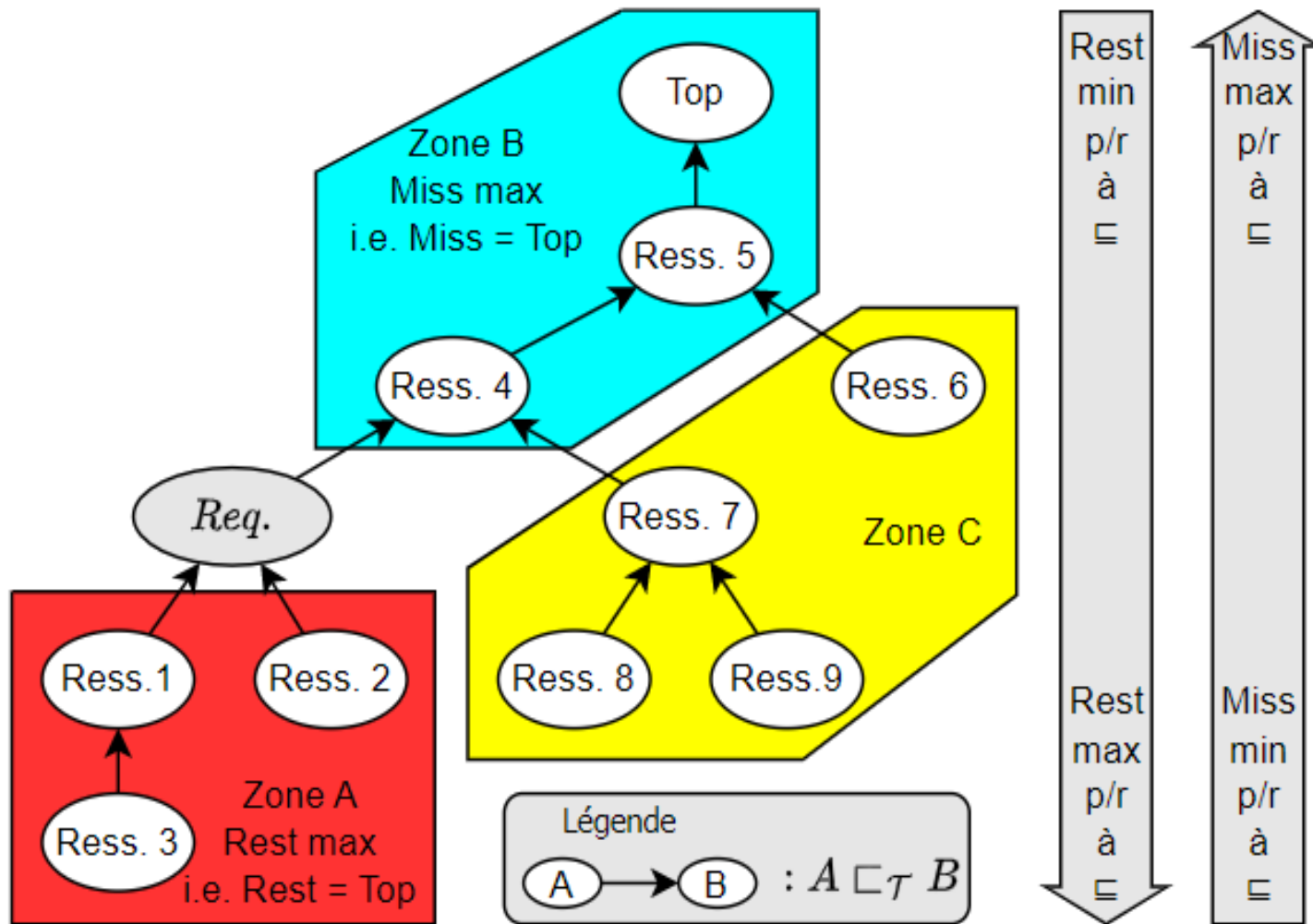
- Ma thèse et travaux ultérieurs
- Thèse A. Mascaro (2019-2023), projet STAM (métrologie) - Collaboration **Perfect Memory** - Financement FEDER

Principe de matchmaking

- Minimiser le Rest d'abord puis le Miss, par rapport à la subsumption, puis par rapport à la taille
- Meilleures couvertures :
 - couvrir au mieux la requête par des ensembles de ressources
- Matchmaking multicritère :
 - Comparer la requête avec chaque ressource via Rest et Miss
 - Et envisager plusieurs composantes à la requête (i.e. plusieurs sous-requêtes)



Matchmaking multicritère



Optimiser
- d'abord le Rest
- puis le Miss

Selon le Rest puis le Miss
N°1 : Offres 1 et 2
N°2 : Offre 3
N°3 : Offres 6, 8 et 9
N°4 : Offre 7
N°5 : Offre 4
N°6 : Offre 5

On a:
 $O_2 \subseteq_T O_1$ implique
 $Rest_{Dem}(O_1) \subseteq_T Rest_{Dem}(O_2)$
et
 $Miss_{Dem}(O_2) \subseteq_T Miss_{Dem}(O_1)$

Selon le Rest uniquement :

- Offres 1, 2 et 3
- Offres 6, 8 et 9
- Offre 7
- Offre 4
- Offre 5

Selon le Miss uniquement :

- Offres 4 et 5
- Offres 1, 2, 6 et 7
- Offres 3, 8 et 9

Hypothèses :

- l'absence de flèche entre 2 offres signifie qu'elles sont incomparables par rapport à \subseteq_T
- les Rest des offres 4 à 9 sont tous non équivalents 2 à 2,
- les Miss des offres 4 à 9 sont tous non équivalents 2 à 2 (sauf ceux des offres 4 et 5 qui valent \top)
- les offres qui sont incomparables 2 à 2 par rapport à \subseteq_T ont des Rest et des Miss qui sont incomparables 2 à 2 par rapport à \subseteq_T

Résultats

- Différence

- Cas des TBox acycliques
 - Définition du problème
 - Algorithmes polynomiaux en les tailles des concepts (ALN et EL)
 - Implémentations
- Cas des TBox cycliques (EL)
 - Instanciation de la définition
 - Caractérisation de la subsomption dans EL avec TBox générales et potentiellement cycliques grâce à des hypergraphes orientés
 - A venir
 - Subsomption caractérisée à partir d'hypergraphes orientés
 - Caractérisation de la différence en termes d'hypergraphes orientés
 - Proposition d'un algorithme de calcul

- Matchmaking

- TBox acycliques seulement
- Meilleures couvertures
 - ALN
 - Algorithme exponentiel
 - Importance du problème des transversaux minimaux d'un hypergraphe
- Matchmaking multicritère
 - EL
 - Algorithme polynomial en la taille de l'expansion complète de la TBox (i.e. exponentiel en la taille de la TBox)
- Implémentations

Rappels sur l'intégration de données et l'OBDA

Rappels sur l'intégration de données

- Problème
 - Des sources de données hétérogènes dans un même domaine
 - Comment interroger facilement toutes ces sources à partir d'un unique schéma global ?
- Exemple dans le domaine des réseaux sociaux
 - Source SN1
 - Source SN2
 - Requête : triplets (id, statut, nb de vues) des utilisateurs de SN1

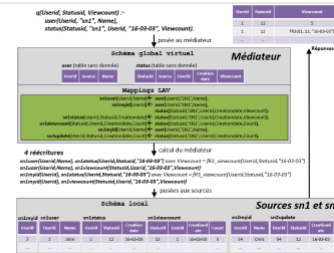
Résultat attendu

id	statut	nombre de vues
1	12	5

Approches d'intégration de données

- Approche matérialisée (entrepôt de données) :
 - On construit une base de données (un entrepôt) avec un schéma global permettant de couvrir les données de toutes les sources.
 - On importe dans l'entrepôt toutes les données de toutes les sources (processus ETL).
 - On pose des requêtes à l'entrepôt.
- Approche virtuelle (médiation) :
 - On construit un schéma global permettant de couvrir les données de toutes les sources.
 - On définit des correspondances (mappings) entre les prédicats du schéma global et les prédicats des sources.
 - On pose des requêtes au schéma global qui sont réécrites en utilisant les prédicats des sources. Les réécritures sont posées aux sources.

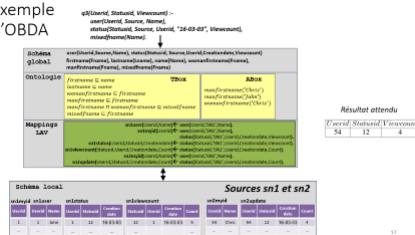
Exemple de médiation



OBDA – Ontology-Based Data Access

- Ajout d'une ontologie au niveau du schéma global
- Intérêt de cette ontologie
 - Pouvoir mémoriser et se servir des connaissances d'un expert
 - Avoir un vocabulaire plus riche pour exprimer des requêtes
 - Améliorer la quantité/pertinence des réponses
- L'ontologie
 - En OWL en général (logiques de description)
 - Parfois ajout de connaissances sous formes de règles aussi
- Ontologie et règles : comment gérer OWA et CWA ?

Exemple d'OBDA



OWA (ontologie) et CWA (règles)

- Exemple en médecine
 - OWA pour raisonner en radiologie ou avec des données de laboratoire
 - « à moins qu'un test radiologique ou de laboratoire ne prouve un fait négatif, aucune hypothèse arbitraire sur ce fait ne peut être avancée »
 - CWA pour raisonner avec des données pharmacologiques : « un patient ne prend pas un traitement particulier si rien ne nous dit qu'il le prend ».
- CWA
 - Tout ce qu'on ne sait pas a priori ou qu'on ne peut pas prouver est considéré comme faux
 - Exemple : si un train n'est pas noté à 10h dans les horaires, c'est qu'il n'y a pas à 10h
- OWA
 - Tout ce qu'on ne sait pas a priori ou qu'on ne peut pas prouver est considéré comme inconnu.
 - Exemple : Je ne sais pas si le facteur est passé. Mais cela ne prouve pas qu'il n'est pas passé.

Rappels sur l'intégration de données

- Problème
 - Des sources de données hétérogènes dans un même domaine
 - Comment interroger facilement toutes ces sources à partir d'un unique schéma global ?
- Exemple dans le domaine des réseaux sociaux
 - Source SN1
 - Source SN2
 - Requête : triplets (id, statut, nb de vues) des utilisateurs de SN1

sn1myid		sn1user		sn1status		sn1viewcount				
userid		userid	name	userid	statusid	creation date	statusid	userid	creation date	count
1		1	John	1	12	16-03-03	12	1	16-03-03	5
...	

sn2myid		sn2update			
userid	name	userid	statusid	creation date	count
54	Chris	54	12	16-03-03	4
...

Résultat attendu

id	statut	nombre de vues
1	12	5

Approches d'intégration de données

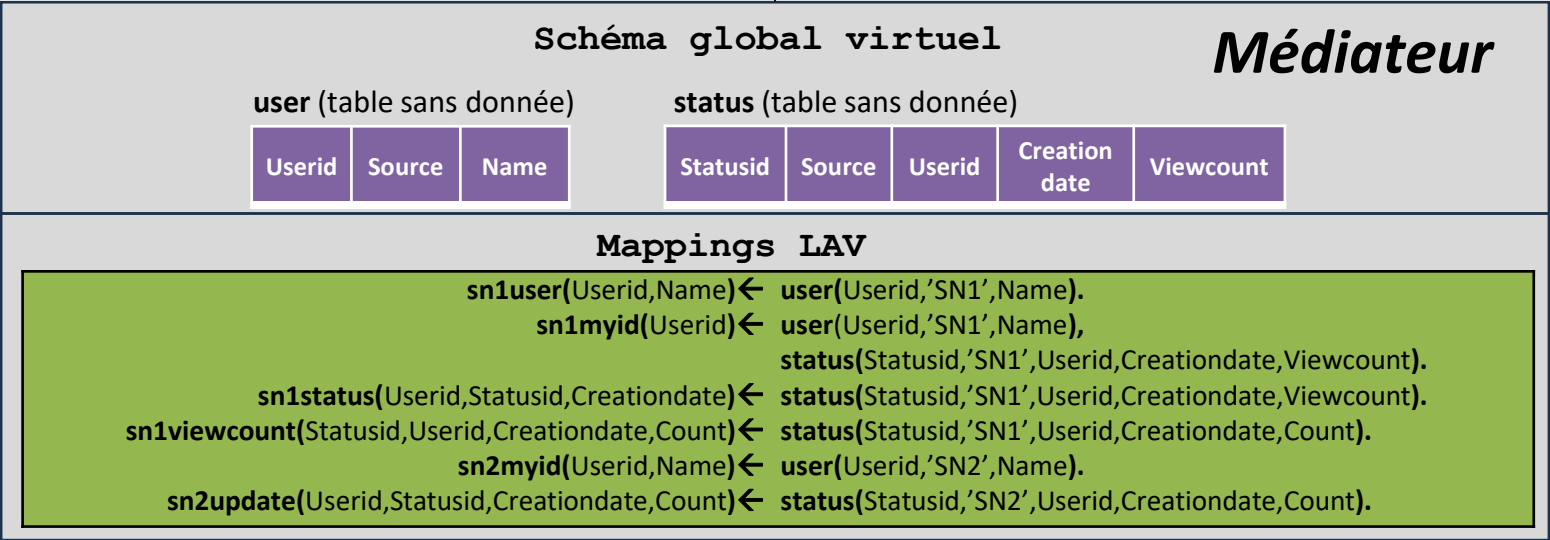
- Approche matérialisée (entrepôts de données) :
 - On construit une base de données (un entrepôt) avec un schéma global permettant de couvrir les données de toutes les sources.
 - On importe dans l'entrepôt toutes les données de toutes les sources (processus ETL).
 - On pose des requêtes à l'entrepôt.
- Approche virtuelle (médiation) :
 - On construit un schéma global permettant de couvrir les données de toutes les sources.
 - On définit des correspondances (mappings) entre les prédicats du schéma global et les prédicats des sources.
 - On pose des requêtes au schéma global qui sont réécrites en utilisant les prédicats des sources. Les réécritures sont posées aux sources.

Exemple de médiation

*q(Userid, Statusid, Viewcount) :-
 user(Userid, "sn1", Name),
 status(Statusid, "sn1", Userid, "16-03-03", Viewcount).*

↓ posée au médiateur

Userid	Statusid	Viewcount
1	12	5
1	12	fR3d(1, 12, "16-03-03")
...



OBDA – Ontology-Based Data Access

- Ajout d'une ontologie au niveau du schéma global
- Intérêt de cette ontologie
 - Pouvoir mémoriser et se servir des connaissances d'un expert
 - Avoir un vocabulaire plus riche pour exprimer des requêtes
 - Améliorer la quantité/pertinence des réponses
- L'ontologie
 - En OWL en général (logiques de description)
 - Parfois ajout de connaissances sous formes de règles aussi
- Ontologie et règles : comment gérer OWA et CWA ?

Exemple d'OBDA

```
q3(Userid, Statusid, Viewcount) :-  
  user(Userid, Source, Name),  
  status(Statusid, Source, Userid, "16-03-03", Viewcount),  
  mixedfname(Name).
```



Schéma global	$user(UserId, Source, Name), status(Statusid, Source, UserId, Creationdate, Viewcount)$ $firstname(Fname), lastname(Lname), name(Name), womanfirstname(Fname),$ $manfirstname(Fname), mixedfname(Fname)$	
Ontologie	$firstname \sqsubseteq name$ $lastname \sqsubseteq name$ $womanfirstname \sqsubseteq firstname$ $manfirstname \sqsubseteq firstname$ $manfirstname \sqcap womanfirstname \sqsubseteq mixedfname$ $mixedfname \sqsubseteq firstname$	TBox ABox $manfirstname('Chris')$ $manfirstname('John')$ $womanfirstname('Chris')$
Mappings LAV	$sn1user(Userid, Name) \leftarrow user(Userid, 'SN1', Name).$ $sn1myid(Userid) \leftarrow user(Userid, 'SN1', Name),$ $status(Statusid, 'SN1', UserId, Creationdate, Viewcount).$ $sn1status(Userid, Statusid, Creationdate) \leftarrow status(Statusid, 'SN1', UserId, Creationdate, Viewcount).$ $sn1viewcount(Statusid, UserId, Creationdate, Count) \leftarrow status(Statusid, 'SN1', UserId, Creationdate, Count).$ $sn2myid(Userid, Name) \leftarrow user(Userid, 'SN2', Name).$ $sn2update(Userid, Statusid, Creationdate, Count) \leftarrow status(Statusid, 'SN2', UserId, Creationdate, Count).$	

Résultat attendu

<i>Usrid</i>	<i>Statusid</i>	<i>Viewcount</i>
54	12	4

[illegible]

OWA (ontologie) et CWA (règles)

Gomes AS, Alferes J, Swift T. A Goal-Directed Implementation of Query Answering for Hybrid MKNF Knowledge Bases. Theory and Practice of Logic Programming. 2014 03;14.

- Exemple en médecine
 - OWA pour raisonner en radiology ou avec des données de laboratoire
« à moins qu'un test radiologique ou de laboratoire ne prouve un fait négatif, aucune hypothèse arbitraire sur ce fait ne peut être avancée »
 - CWA pour raisonner avec des données pharmacologiques : « un patient ne prend pas un traitement particulier si rien ne nous dit qu'il le prend ».
- CWA
 - Tout ce qu'on ne sait pas a priori ou qu'on ne peut pas prouver est considéré comme faux
 - Exemple : si un train n'est pas noté à 10h dans les horaires, c'est qu'il n'y en a pas à 10h
- OWA
 - Tout ce qu'on ne sait pas a priori ou qu'on ne peut pas prouver est considéré comme inconnu.
 - Exemple : Je ne sais pas si le facteur est passé. Mais cela ne prouve pas qu'il n'est pas passé.

Intégration de données déclarative par réécriture de requêtes

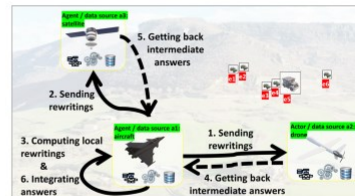
Alimentation semi-automatique d'un entrepôt de données à partir d'une ontologie datalog de services web

- Utiliser une approche virtuelle d'intégration de données pour faciliter l'ETL dans une approche matérialisée d'intégration de données.
- Réalisation de l'intégration de données
 - Adaptation de l'algorithme inverse-rules (gestion de dépendances en plus des mappings)
 - Les sources sont des API de services web (avec des entrées et des sorties).
- Génie logiciel déclaratif : chaque étape d'intégration est
 - Soit automatisée
 - Soit paramétrée à la main via un langage déclaratif (requêtes conjonctives, datalog pour les mappings, XSD pour la validation des réponses, XSLT pour la transformation des réponses)→ on passe du développement au paramétrage

20

Intégration de données distribuées par SLG-résolution avec une ontologie

- Domaine de l'aéronautique militaire (coalition d'agents en mission)
- Particularités du cas d'étude
 - Les sources sont distribuées c'est-à-dire pas accessibles tout le temps.
 - Besoin d'ajouter des règles en plus de l'ontologie
- Réécriture de requête adaptée à des données distribuées ?
- Performances ?
- Solution proposée
 - Traduction de l'ontologie en règles (profil OWL peu expressif donc pas de soucis de sémantique) et faits
 - Réécriture de requêtes par SLG-résolution (décidable si pas de fonction, pas de calculs redondants)
- Implémentation avec JENA



21

- Thèse de John Samuel (2011-2014) – Collaboration **Rootsystem** – Financement Région Auvergne et FEDER
- Postdoctorat Karima Ennaoui (2019-2021), Ingénieur d'études Mathieu Faivre (2019-2020) – Collaboration **Dassault Aviation** – Financement DGA

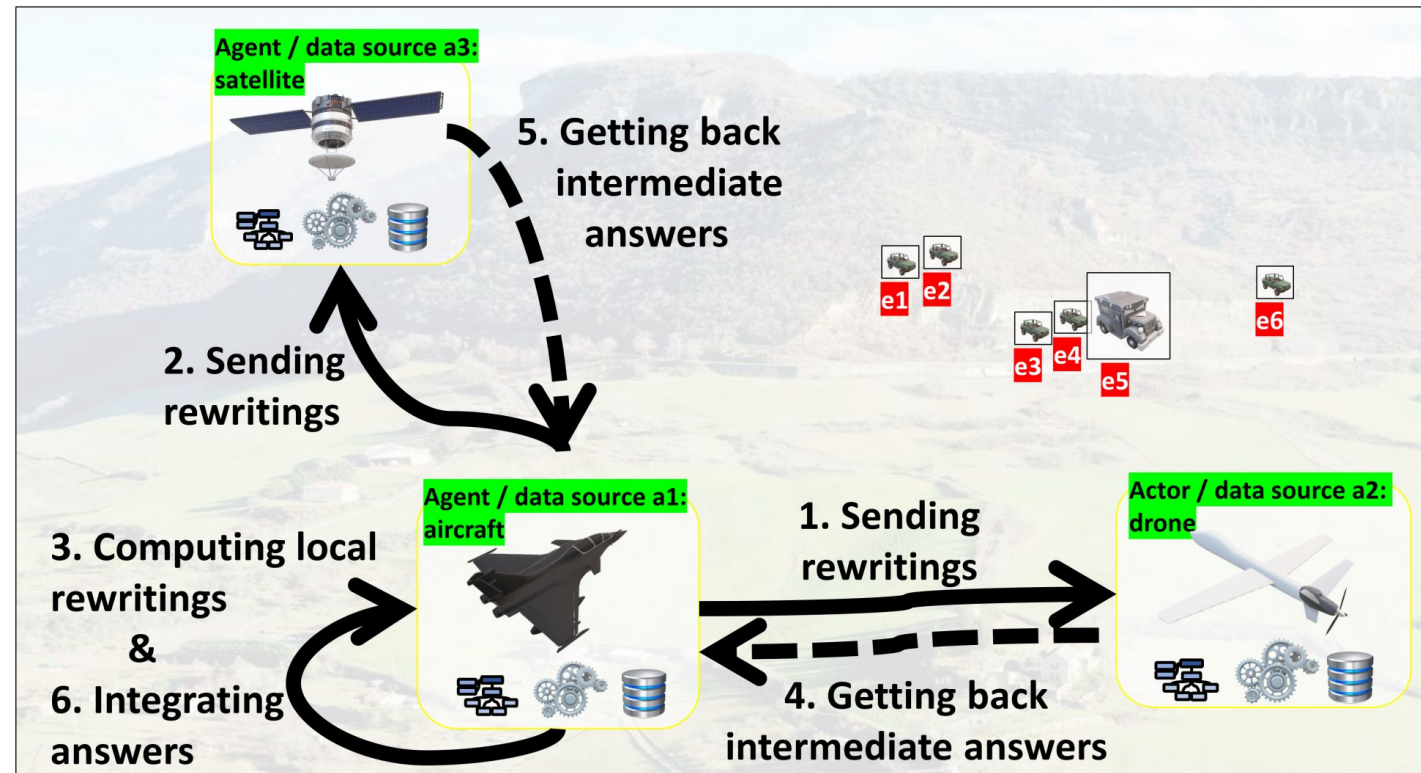
Alimentation semi-automatique d'un entrepôt de données à partir d'une ontologie datalog de services web

- Utiliser une approche virtuelle d'intégration de données pour faciliter l'ETL dans une approche matérialisée d'intégration de données.
- Réalisation de l'intégration de données
 - Adaptation de l'algorithme inverse-rules (gestion de dépendances en plus des mappings)
 - Les sources sont des API de services web (avec des entrées et des sorties).
- Génie logiciel déclaratif : chaque étape d'intégration est
 - Soit automatisée
 - Soit paramétrée à la main via un langage déclaratif (requêtes conjonctives, datalog pour les mappings, XSD pour la validation des réponses, XSLT pour la transformation des réponses)

➔ on passe du développement au paramétrage

Intégration de données distribuées par SLG-résolution avec une ontologie

- Domaine de l'aéronautique militaire (coalition d'agents en mission)
- Particularités du cas d'étude
 - Les sources sont distribuées c'est-à-dire pas accessibles tout le temps. Besoin d'ajouter des règles en plus de l'ontologie
- Réécriture de requête adaptée à des données distribuées ?
- Performances ?
- Solution proposée
 - Traduction de l'ontologie en règles (profil OWL peu expressif donc pas de soucis de sémantique) et faits
 - Réécriture de requêtes par SLG-résolution (décidable si pas de fonction, pas de calculs redondants)
- Implémentation avec JENA



Recommandation et prise de décision déclarative à l'aide d'ontologies et de règles

Recommandation de défis physiques dans une application de e-santé

- **Projet Oraloos de Jeolis Solution**
- **Application de e-santé** : proposer régulièrement des défis physiques à des patients afin qu'ils maintiennent leurs efforts pour lutter contre leur surpoids
- **Détermination du prochain défi à proposer en fonction**
 - D'une ontologie des défis et des patients (partie OWL)
 - De règles basées sur les résultats (réussites ou échecs) aux précédents défis et du feedback du patient
- **Enjeux**
 - Raisonner avec des ontologies et des règles
 - Proposer une méthodologie de développement déclarative (pas besoin de coder pour ajouter des connaissances)
- **Plusieurs versions**
 - V1 : OWL + SWRL : traitement Python de la négation
 - V2 : OWL + ASP via HexLite : ça marche, mais mise en œuvre pas très facile.
 - V3 : OWL + ASP via la construction d'une bibliothèque Exalis. Tests en cours
- **Développement en python**
 - Accès aux ontologies : OWLready2
 - Raisonneur OWL : Hermit
 - ASP solveur permettant l'appel au raisonneur OWL : Hesite (v2) ou Clingo (v3)
- **Premiers résultats qualitativement bons.**

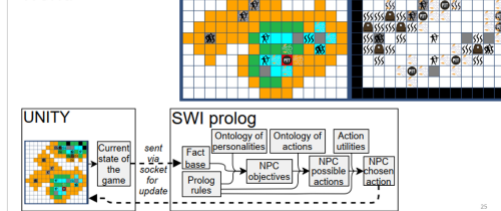
23

Prise de décision pour des personnages non joueurs dans les jeux vidéo

- **Enjeux**
 - Proposer des IA de PNJ basées sur des raisonnements logiques (et plus des ensembles de tests, cf. les FSM, les arbres de comportement)
 - Proposer une méthodologie déclarative
 - Prise en main par les game designers (via une interface adaptée)
 - Possibilité de passer à l'échelle (ajouter beaucoup de connaissances sans que le système ne devienne ingérable)
 - Identifier les étapes bloquantes pour l'exécution en temps réel
- **Architecture proposée**
 - Un programme prolog qui orchestre par SLG-résolution les règles de prise de décision principales avec un dialogue avec des ontologies
 - Des ontologies OWL du domaine (éléments de situation, traits de caractère, émotions, actions, ...)
 - Un design pattern utilisant la WFS (well founded semantics) pour représenter la connaissance partielle des PNJ
- **Particularité de l'intégration règles et ontologie** : les prédicats de l'ontologie sont des constantes pour le programme prolog (et non des prédicats)

24

Exemple du jeu Wumpus World



- Ingénieur d'études Adrien Collange (2022-2024) - Collaboration **Jeolis Solutions** – financement ANR Plan de relance
- Thèse de Syvain Lapeyrade (2020-...) - Collaboration **Wako Factory** – Financement ANR Contrat doctoral en IA

Recommandation de défis physiques dans une application de e-santé

- Projet Oraloos de Jeolis Solution
- Application de e-santé : proposer régulièrement des défis physiques à des patients afin qu'ils maintiennent leurs efforts pour lutter contre leur surpoids
- Détermination du prochain défi à proposer en fonction
 - D'une ontologie des défis et des patients (partie OWL)
 - De règles basées sur les résultats (réussites ou échecs) aux précédents défis et du feedback du patient
- Enjeux
 - Reasonner avec des ontologies et des règles
 - Proposer une méthodologie de développement déclarative (pas besoin de coder pour ajouter des connaissances)
- Plusieurs versions
 - V1 : OWL + SWRL : traitement Python de la négation
 - V2 : OWL + ASP via HexLite : ça marche, mais mise en œuvre pas très facile.
 - V3 : OWL + ASP via la construction d'une bibliothèque Exialis. Tests en cours
- Développement en python
 - Accès aux ontologies : OWLready2
 - Reasonneur OWL : HermiT
 - ASP solveur permettant l'appel au reasonneur OWL : HexLite (v2) ou Clingo (v3)
- Premiers résultats qualitativement bons.

Prise de décision pour des personnages non joueurs dans les jeux vidéo

- Enjeux

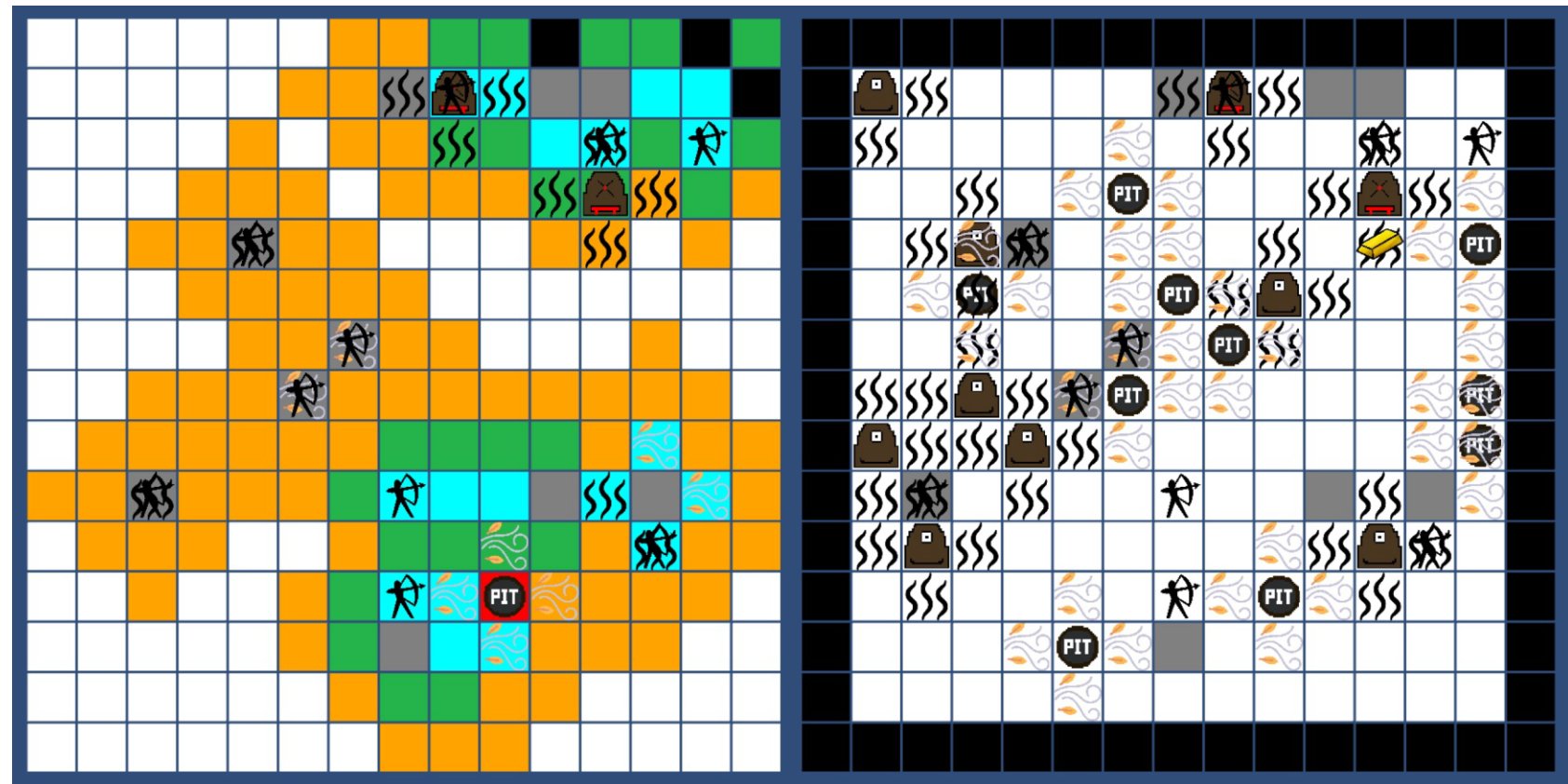
- Proposer des IA de PNJ basées sur des raisonnements logiques (et plus des ensembles de tests, cf. les FSM, les arbres de comportement)
- Proposer une méthodologie déclarative
 - Prise en main par les game designers (via une interface adaptée)
 - Possibilité de passer à l'échelle (ajouter beaucoup de connaissances sans que le système ne devienne ingérable)
- Identifier les étapes bloquantes pour l'exécution en temps réel

- Architecture proposée

- Un programme prolog qui orchestre par SLG-résolution les règles de prise de décision principales avec un dialogue avec des ontologies
- Des ontologies OWL du domaine (éléments de situation, traits de caractère, émotions, actions, ...)
- Un design pattern utilisant la WFS (well founded semantics) pour représenter la connaissance partielle des PNJ

- Particularité de l'intégration règles et ontologie : les prédicats de l'ontologie sont des constantes pour le programme prolog (et non des prédicats)

Exemple du jeu Wumpus World



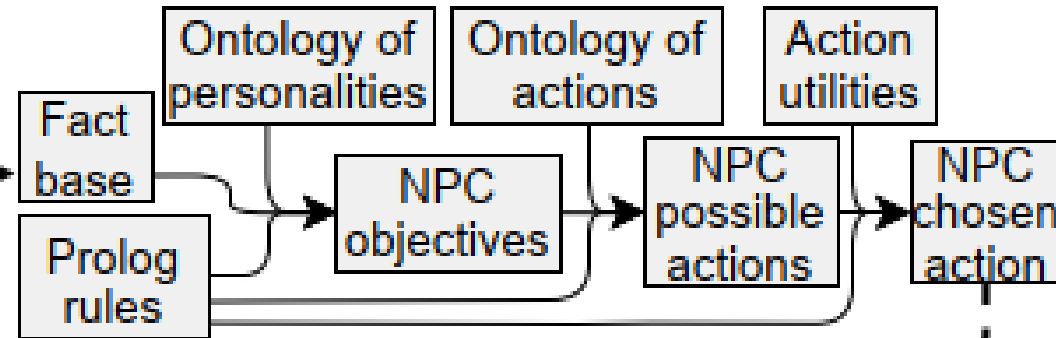
UNITY



Current
state of
the
game

sent
via
socket
for
update

SWI prolog



Travaux en cours

- Travaux en cours
 - Finir le travail sur la différence avec des TBox cycliques
 - Travail sur la sémantique bien fondée en programmation logique
 - Autre caractérisation que par point fixe
 - Etude théorique du pattern proposé en prise de décision
 - Possibilité de raisonner en OWA et en CWA ?
 - Poursuite de l'exploration des approches hybrides ontologies et règles
- Applications en
 - Aéronautique (Thales, thèse CIFRE de Arun Raveendran Nair Sheela)
 - E-santé (Jeolis Solutions)
 - Game AI
- Extension au neuro-symbolique ?

Merci !