

Un système multi-agent adaptatif  
pour  
la construction d'ontologies à partir de textes  
Soutenance de thèse

Kévin Ottens  
IRIT - Université Paul Sabatier  
2 octobre 2007

# Titre de cette thèse

## Construction d'Ontologies

- Domaine montant de l'Ingénierie des Connaissances
- Nombreuses utilisations différentes des résultats
  - Plus ou moins formelles
  - Plus ou moins d'éléments terminologiques

## A partir de Textes

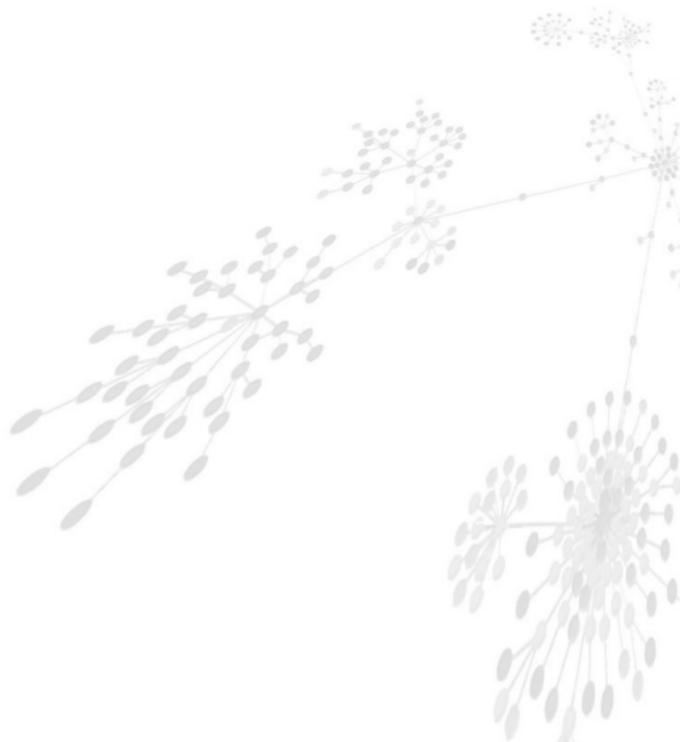
- Implique l'utilisation
  - D'outils de Traitement Automatique de la Langue
  - D'une approche basée sur des corpus textuels

## Système Multi-Agent Adaptatif pour...

- Construction d'ontologies distribuée et dynamique pour prévoir la maintenance
- Principalement intéressant pour le côté ingénierie

# Plan

- 1 Contexte
- 2 Contributions
- 3 Application
- 4 Perspectives
- 5 Conclusion



# Plan

- 1 Contexte
  - Vers les ontologies dynamiques
  - Des systèmes multi-agents
- 2 Contributions
- 3 Application
- 4 Perspectives
- 5 Conclusion



# Plan

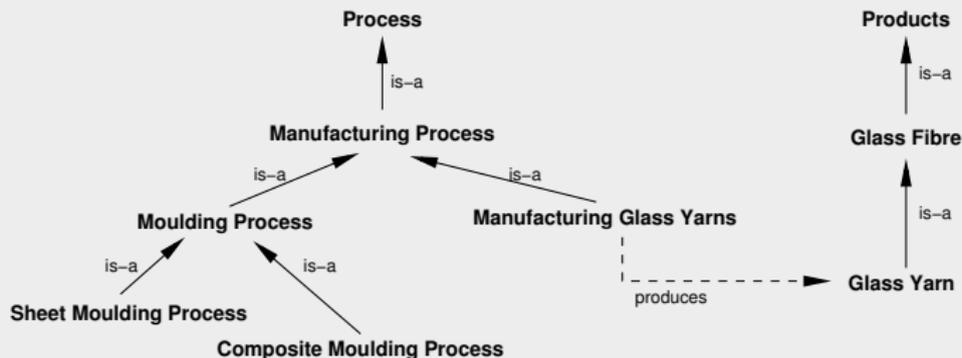
- 1 Contexte
  - Vers les ontologies dynamiques
  - Des systèmes multi-agents
- 2 Contributions
- 3 Application
- 4 Perspectives
- 5 Conclusion



# Ontologies : Définitions & Motivations

## Définitions

- Description structurée du vocabulaire d'un domaine [Bachimont 04]
- Ensemble de concepts structuré par des relations sémantiques [Charlet 03]
- Réseau sémantique dont les concepts sont définis de manière consensuelle



# Ontologies : Définitions & Motivations

## Définitions

- Description structurée du vocabulaire d'un domaine [Bachimont 04]
- Ensemble de concepts structuré par des relations sémantiques [Charlet 03]
- Réseau sémantique dont les concepts sont définis de manière consensuelle

## Motivations

- Améliorer le partage de connaissances entre personnes [Slodzian 00]
- Assurer l'interopérabilité entre applications à base de connaissances [Gruber 91]
- Permettre la réutilisation de modèles de connaissances [Gruber 91]
- Faciliter la communication entre agents logiciels [Bergenti 02]
- Améliorer la recherche d'informations [Baziz 05]

# Construction à partir de textes

## Outillage

### Différents types

- Identification de termes (ANA, Syntex, Termostat...)
- Identification de classes sémantiques (LexiClass, Asium...)
- Identification de relations lexicales et sémantiques (Prométhée, Caméléon...)

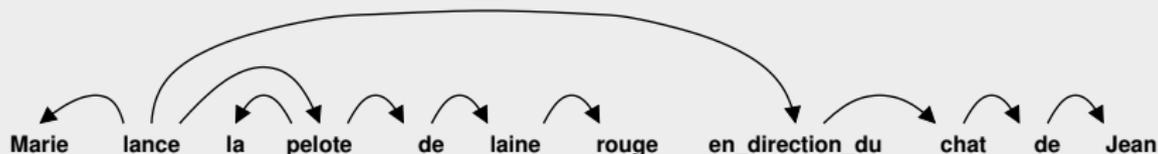
### De l'interprétation

- Profusion de résultats [Assadi 98, Bourigault 03]
- Résultats au niveau linguistique
- Requier une validation de l'ontologue pour donner le statut conceptuel recherché [Aussenac 02, Séguéla 01]

# Syntax [Bourigault 07]

## Principe

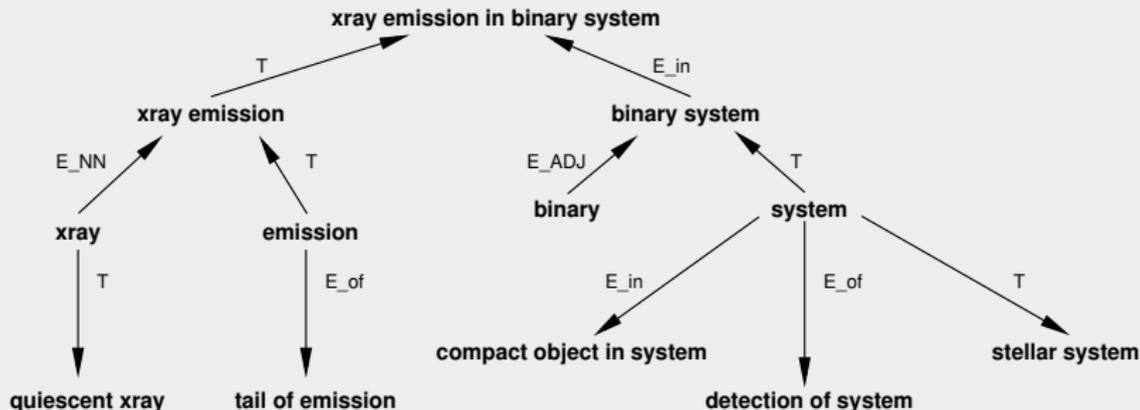
- Etiquetage morpho-syntaxique préalable
- Propositions de dépendances syntaxiques pour chaque phrase
- Extraction d'un réseau de termes



## Principe

- Etiquetage morpho-syntaxique préalable
- Propositions de dépendances syntaxiques pour chaque phrase
- Extraction d'un réseau de termes

## Réseau Tête-Expansion



## Principe

- Algorithme de classification
- Exploration du réseau Tête-Expansion
  - Détermine des contextes partagés
  - Permet le calcul d'un indice de similarité

## Principe

- Algorithme de classification
- Exploration du réseau Tête-Expansion
  - Détermine des contextes partagés
  - Permet le calcul d'un indice de similarité

|           | ADMISSIBLE | NOMINAL | HT | THT | 63KV |
|-----------|------------|---------|----|-----|------|
| CÂBLE     | 0          | 0       | 1  | 0   | 1    |
| INTENSITÉ | 1          | 1       | 0  | 0   | 0    |
| LIGNE     | 0          | 0       | 1  | 1   | 1    |
| TENSION   | 1          | 1       | 0  | 0   | 0    |

## Principe

- Algorithme de classification
- Exploration du réseau Tête-Expansion
  - Détermine des contextes partagés
  - Permet le calcul d'un indice de similarité

|              | ADMISSIBLE | NOMINAL | HT       | THT      | 63KV     |
|--------------|------------|---------|----------|----------|----------|
| <b>CÂBLE</b> | 0          | 0       | <b>1</b> | 0        | <b>1</b> |
| INTENSITÉ    | 1          | 1       | 0        | 0        | 0        |
| <b>LIGNE</b> | 0          | 0       | <b>1</b> | <b>1</b> | <b>1</b> |
| TENSION      | 1          | 1       | 0        | 0        | 0        |

# Construction à partir de textes

## Limites

### Efforts [Aussenac 03]

- Trop d'informations brutes extraites
- Structures obtenues moins formalisées (mais plus riches sur le plan terminologique)
- Les textes ne contiennent pas toute la connaissance

### Maintenance

- Ontologies conçues pour être figées, mais les domaines à modéliser sont en évolution [Aussenac 03]

→ *Ontologies dynamiques* [Maedche & Staab]

# Construction à partir de textes

## Limites

### Efforts [Aussenac 03]

- Trop d'informations brutes extraites
- Structures obtenues moins formalisées (mais plus riches sur le plan terminologique)
- Les textes ne contiennent pas toute la connaissance

### Maintenance

- Ontologies conçues pour être figées, mais les domaines à modéliser sont en évolution [Aussenac 03]

→ *Ontologies dynamiques* [Maedche & Staab]

# Positionnement

## Constats

- Corpus riches comme source d'information
- L'ontologue a un rôle capital pour interpréter les résultats des outils de TAL
- Manque de souplesse dans la production d'ontologies
- Structure hiérarchique comme ossature principale de l'ontologie

## Conséquences

- Corpus textuel comme source première d'information
- Le système doit
  - Faire des propositions modifiables par l'ontologue
  - Permettre de faire évoluer l'ontologie
- Solution basée sur la classification

# Plan

- 1 Contexte
  - Vers les ontologies dynamiques
  - Des systèmes multi-agents
- 2 Contributions
- 3 Application
- 4 Perspectives
- 5 Conclusion



# SMA & Ontologies

## Utilisations classiques

- Ontologies pour les systèmes multi-agents
  - Communication entre les agents [Gandon 02]
  - Interprétation d'un flux de données en entrée [Elmore 03]
- Ontologies pour les agents
  - Représentation du monde [Siebes 02]
- Systèmes multi-agents pour les ontologies
  - Alignement distribué d'ontologies [Lister 06]
  - Aide pour la construction en environnement collaboratif [Bao 04]

# SMA & Ontologies

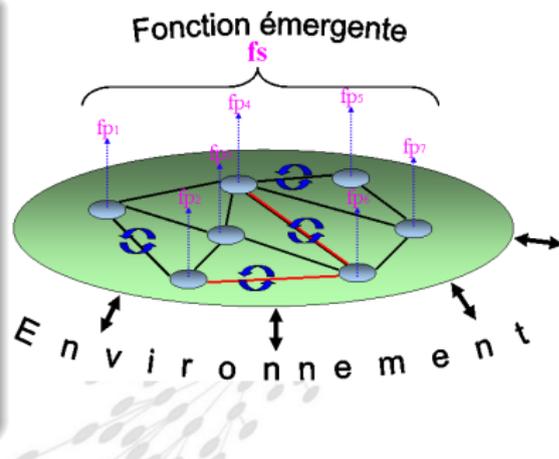
## Nos choix

- Faire le lien entre système multi-agent et construction d'ontologies
  - UML déjà utilisé pour de la modélisation d'ontologies [Cranefield 99]
  - Agents comme évolution des objets [Odell 00]
- Système multi-agent basé sur la classification de termes
  - Utilisation fructueuse de la classification pour la construction d'ontologies [Assadi 98, Faure 00]
  - Un exemple d'algorithme distribué de classification en data-mining [Parunak 06]

# Théorie des Adaptive Multi-Agent Systems [Camps 98]

## AMAS

- Chaque agent produit une fonction  $f_{P_i}$
- Leurs liens engendrent la fonction  $f_S$
- Adaptation par auto-organisation  
→ heuristique



## Coopération

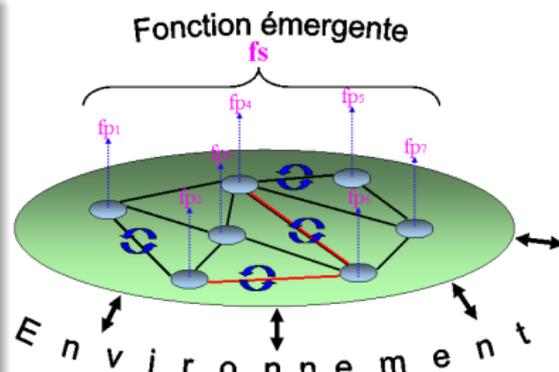
$$\text{NON COOPERATION} = \neg C_1 \vee \neg C_2 \vee \neg C_3$$

- $\neg C_1$  : Incompréhension
- $\neg C_2$  : Défaut de raisonnement
- $\neg C_3$  : Inutilité de l'action

# Théorie des Adaptive Multi-Agent Systems [Camps 98]

## AMAS

- Chaque agent produit une fonction  $f_{P_i}$
- Leurs liens engendrent la fonction  $f_S$
- Adaptation par auto-organisation  
→ heuristique



## Coopération

$$\text{NON COOPERATION} = \neg C_1 \vee \neg C_2 \vee \neg C_3$$

- $\neg C_1$  : Incompréhension
- $\neg C_2$  : Défaut de raisonnement
- $\neg C_3$  : Inutilité de l'action

# ADELFE : Processus d'ingénierie agent [Picard 04]

## Survol

- Processus basé sur Neptune, lui même basé sur le Rational Unified Process
- Ajoute des activités spécifiques à l'ingénierie agent
- Ne donne aucune indication sur le passage entre la conception et l'implémentation



# Positionnement

## Constats

- Pas de SMA pour la construction d'ontologies à partir de textes
- Pas de SMA représentant une ontologie
- Piste pour la classification distribuée
- ADELFE incomplet pour la conception et l'implémentation

## Conséquences

- Grands axes de notre système
  - SMA en lien avec des données de type terminologique
  - Coeur du comportement basé sur un algorithme distribué de classification
- Extension nécessaire d'ADELFE pour s'attaquer au problème

# Plan

## 1 Contexte

## 2 Contributions

- Ingénierie des modèles & SMA
- Dynamo

## 3 Application

## 4 Perspectives

## 5 Conclusion



# Plan

## 1 Contexte

## 2 Contributions

- Ingénierie des modèles & SMA
- Dynamo

## 3 Application

## 4 Perspectives

## 5 Conclusion



# ADELFE et la conception

## Statut d'ADELFE [Picard 05]

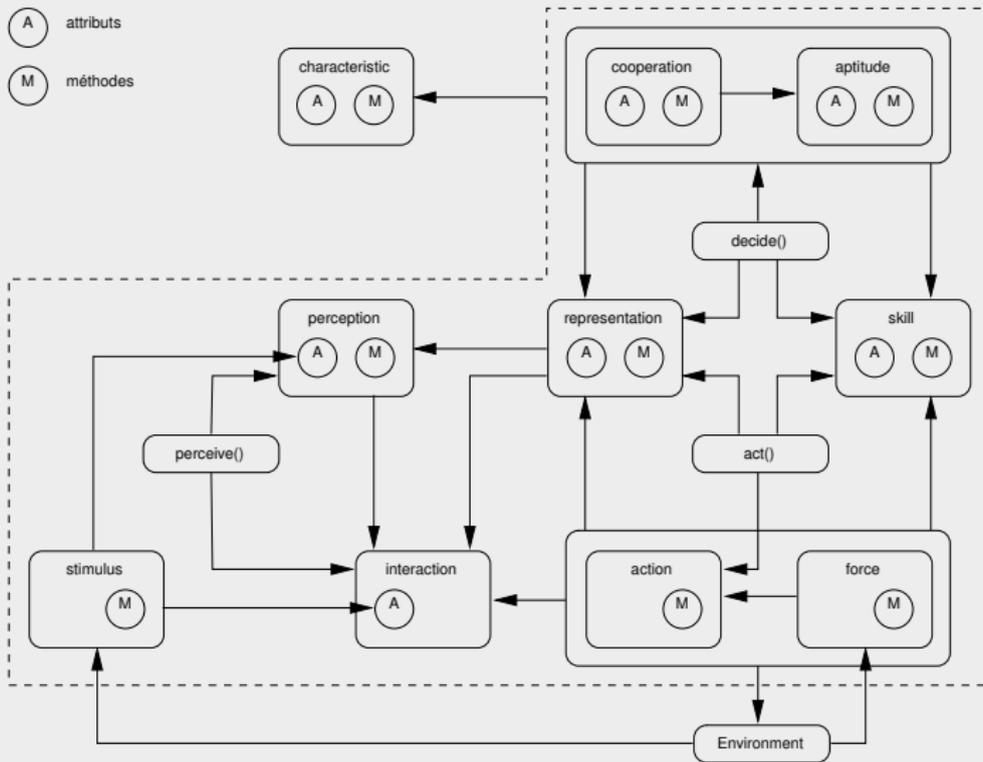
- Utilisation du paradigme objet pour le développement des agents
- Un guide sur l'utilisation des AMAS
- Description des agents par stéréotypage
- Couvre les phases préliminaires du développement, mais pas la conception ou le codage des agents

## Apports nécessaires

- Un système de stéréotypes plus précis pour la conception
- Des mécanismes de génération de code [Kleppe 03]

# Stéréotypes [Ottens 06]

## Aperçu



# Stéréotypes

## Conséquences

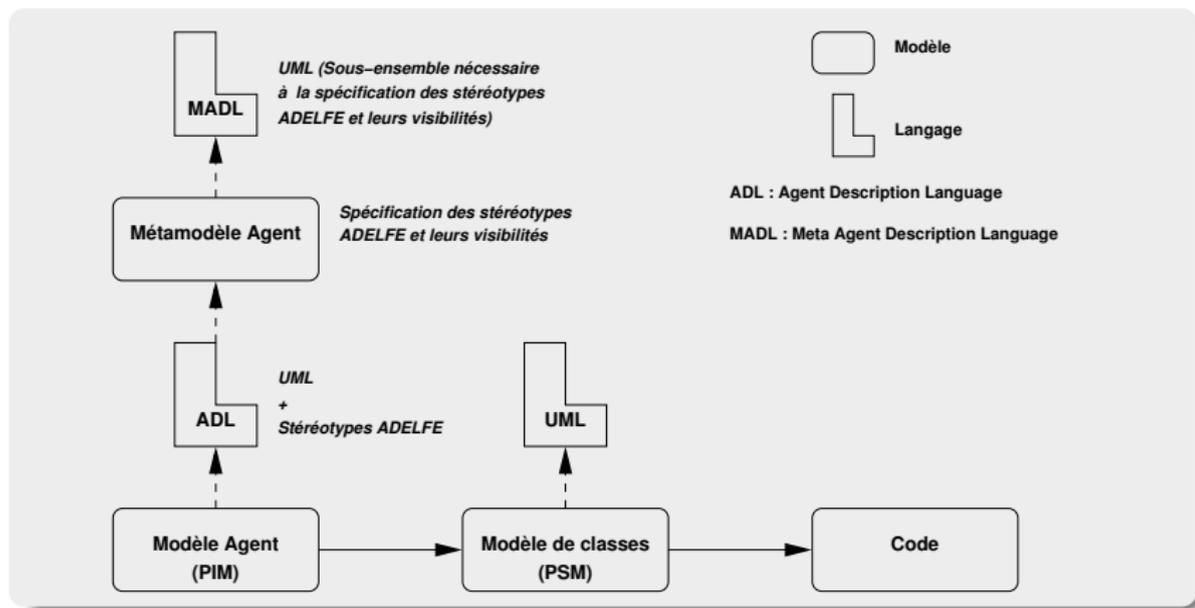
### Conception des agents

- Aspect statique entièrement couvert
- Décrit quelles informations peuvent être manipulées à tout instant

### Assurer le respect de la conception

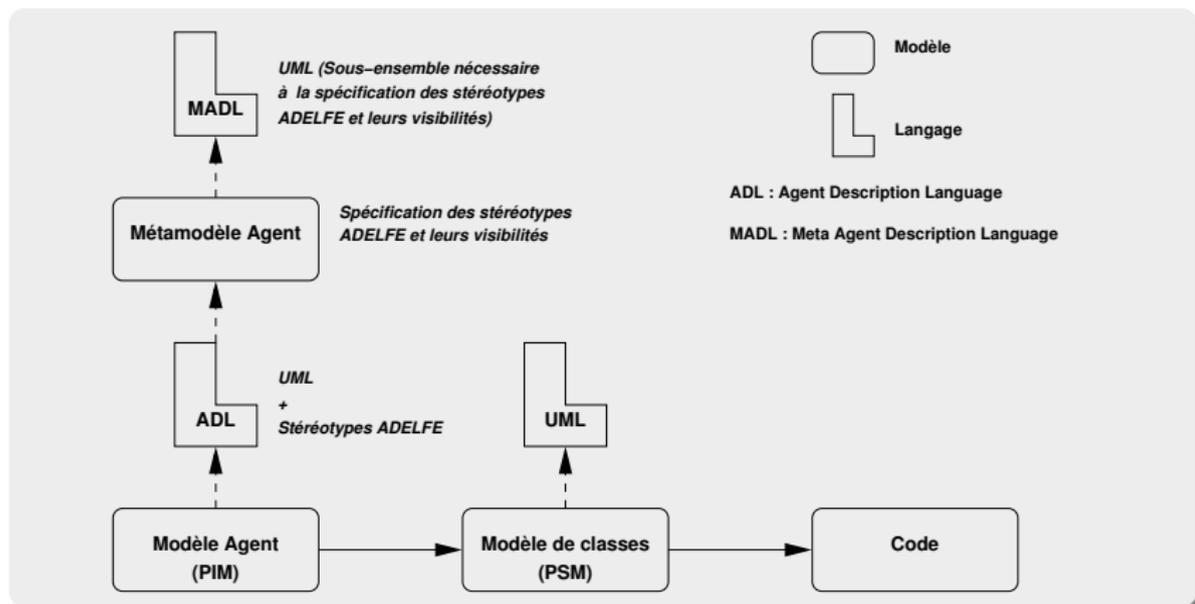
- Patron objet réduisant l'accès aux données
- Chaque stéréotype donne lieu à une classe respectant ce patron
- Ce niveau de description permet la génération de code

## Génération de code [Ottens 06]



ADELFE permet à présent de couvrir la conception et le codage

## Génération de code [Ottens 06]



ADELFE permet à présent de couvrir la conception et le codage

# Plan

## 1 Contexte

## 2 Contributions

- Ingénierie des modèles & SMA
- Dynamo

## 3 Application

## 4 Perspectives

## 5 Conclusion



# Application d'ADELFE

## Expression des besoins finals

- Termes et concepts comme entités essentielles
- Environnement
  - Social
  - Intégrant l'ontologue
  - Sortie de l'extracteur

## Analyse

- Adéquation des AMAS
  - Problème complexe sans solution algorithmique connue
  - Nombre important d'entités pour la résolution
  - Besoin de dynamique et d'adaptation
- Agents-Concepts se positionnant dans la hiérarchie

# Conception : « Le SMA est l'ontologie »

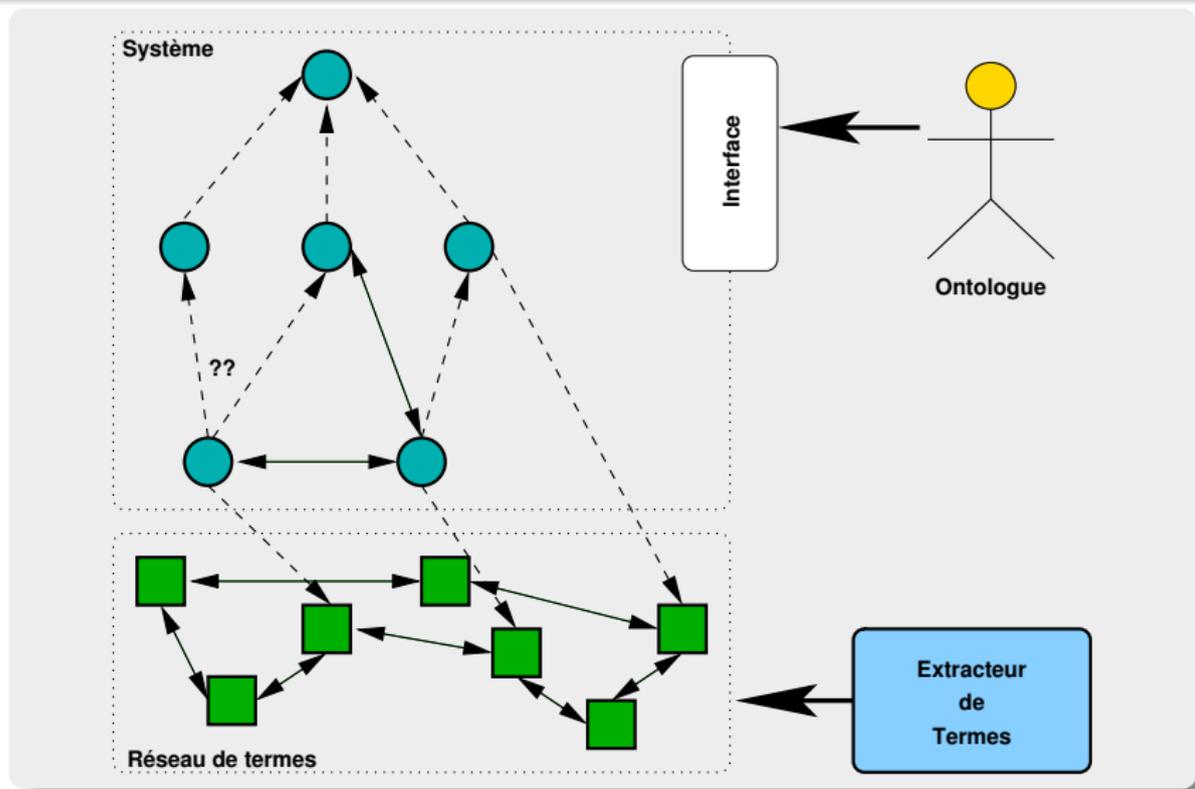
## Vers moins d'intervention manuelle

- Exploiter les informations en provenance de Syntax
- Faire valider une organisation conceptuelle
- Réagir aux modifications de l'ontologue

## Un autre point de vue sur l'ontologie

- Ontologie = équilibre entre les agents-concepts qui la composent
- Modification de l'ontologie = perturbation de l'équilibre
- Ontologie dynamique = processus auto-organisé cherchant à résorber les perturbations

# Architecture



# Structure des agents

## Modèle de nos agents

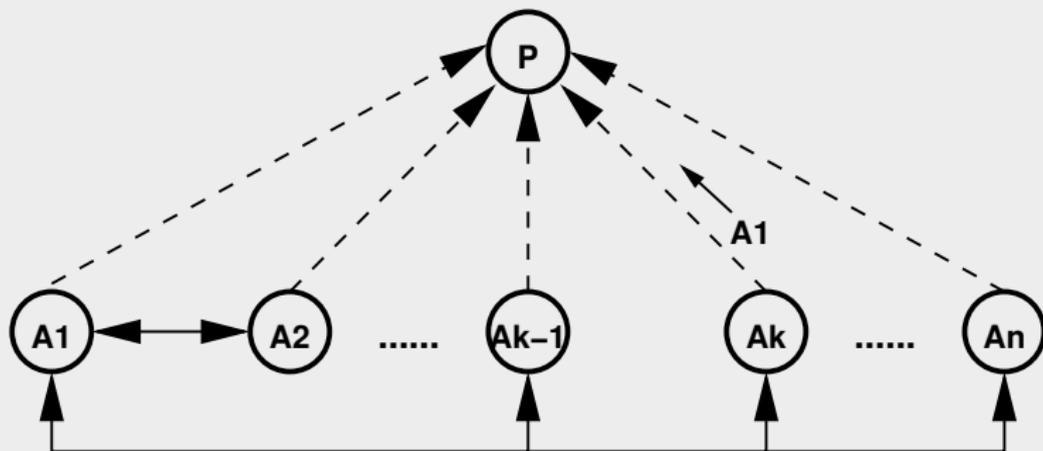
- *interaction* : boîte de messages
- *perception* : extraction d'information des messages reçus
- *representation* : relation avec les autres, terme associé, actions planifiées
- *cooperation* : détection des situations non coopératives, et système de règles condition/action
- *action* : exécution des actions planifiées (communication, gestion des agents, etc.)

## Des modèles

- Notre modèle d'agent s'est construit incrémentalement
- Uniquement la partie comportement à implémenter

# Algorithme distribué de classification [Ottens 07]

## Vue locale

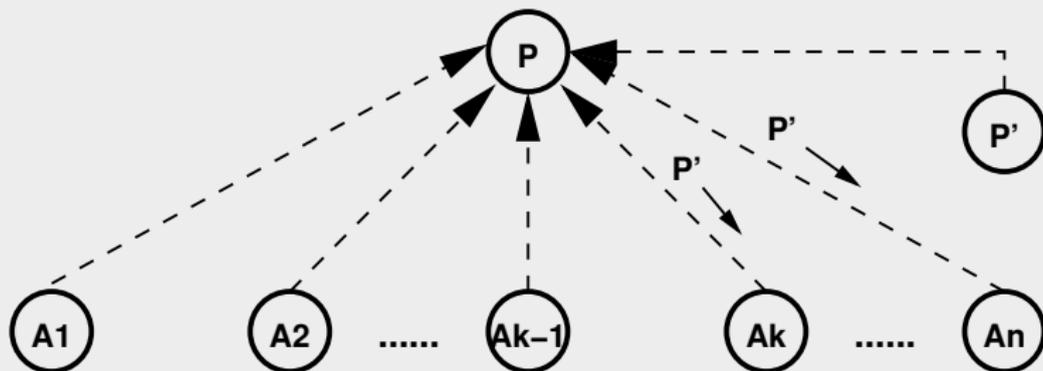


## Étapes

- 1 Évaluation des similarités et « votes »
- 2 Partitionnement et création du niveau intermédiaire
- 3 Changement de parent

# Algorithme distribué de classification [Ottens 07]

## Vue locale

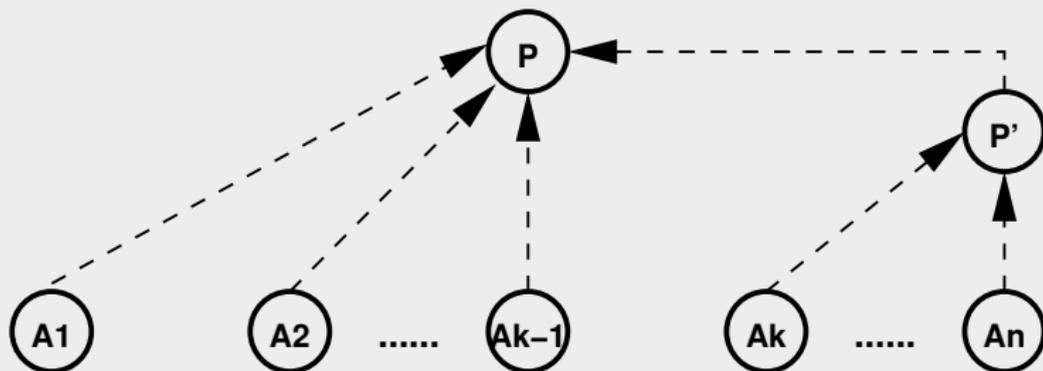


## Etapas

- 1 Évaluation des similarités et « votes »
- 2 Partitionnement et création du niveau intermédiaire
- 3 Changement de parent

# Algorithme distribué de classification [Ottens 07]

## Vue locale

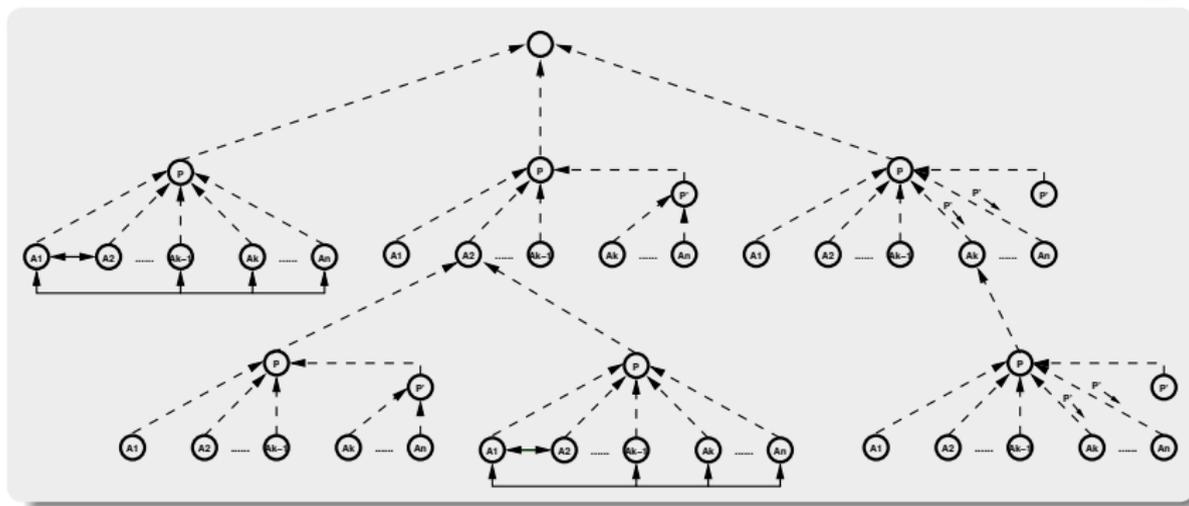


## Etapas

- 1 Évaluation des similarités et « votes »
- 2 Partitionnement et création du niveau intermédiaire
- 3 **Changement de parent**

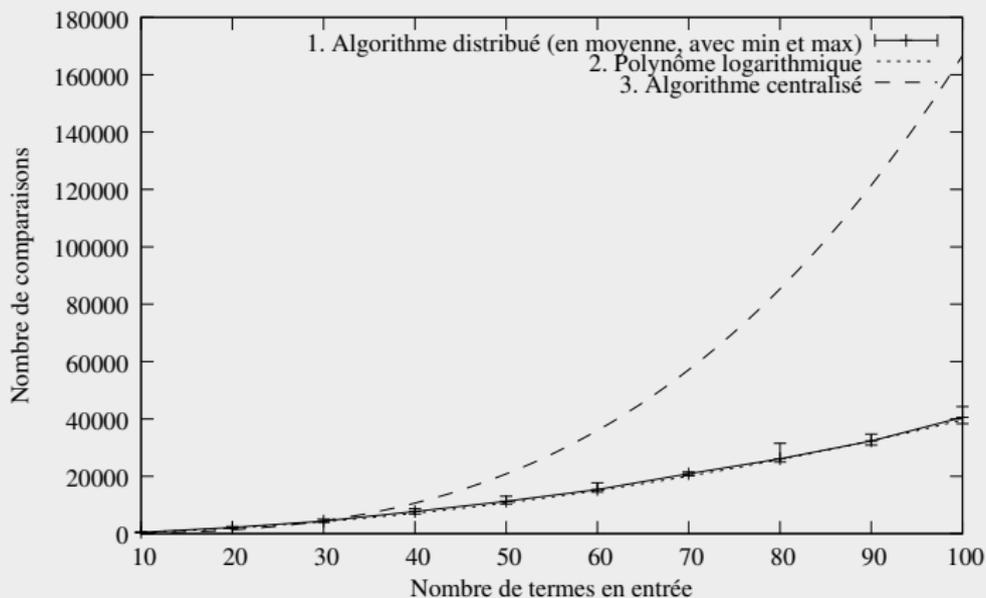
# Algorithme distribué de classification

Vue globale



# Algorithme distribué de classification

## Complexité expérimentale



- Complexité en moyenne:  $O(n^2 \log(n))$
- Écart-type maximum: environ 5%

# Algorithme distribué de classification

## Point de vue qualitatif

### Exécution automatique

- Vue permanente de la hiérarchie en construction
- Permet d'obtenir un « premier brouillon »

### Intervention de l'ontologue

- Pas d'ajustement de l'algorithme requis
- Dynamicité, la structure est révisée

# Règle de couverture en tête [Ottens 07]

## Comportement recherché

### Observations

- Il existe des paires de termes où la similitude n'est pas définie
- Les ontologues ont des heuristiques spécifiques à la structuration des feuilles

### Buts

- Pouvoir placer ces termes
- Implémenter une heuristique similaire

### Fonction d'adéquation du parent

- Le meilleur parent pour  $E$  est l'agent  $P$  qui maximise  $a(P, E)$
- *Quand un agent  $E$  est insatisfait de son père  $P$ , il évalue  $a(F_i, E)$  avec tous ses frères (notés  $F_i$ ). Celui maximisant  $a(F_i, E)$  est alors choisi comme nouveau parent*

# Gérer plusieurs critères

## Guide

### Comment?

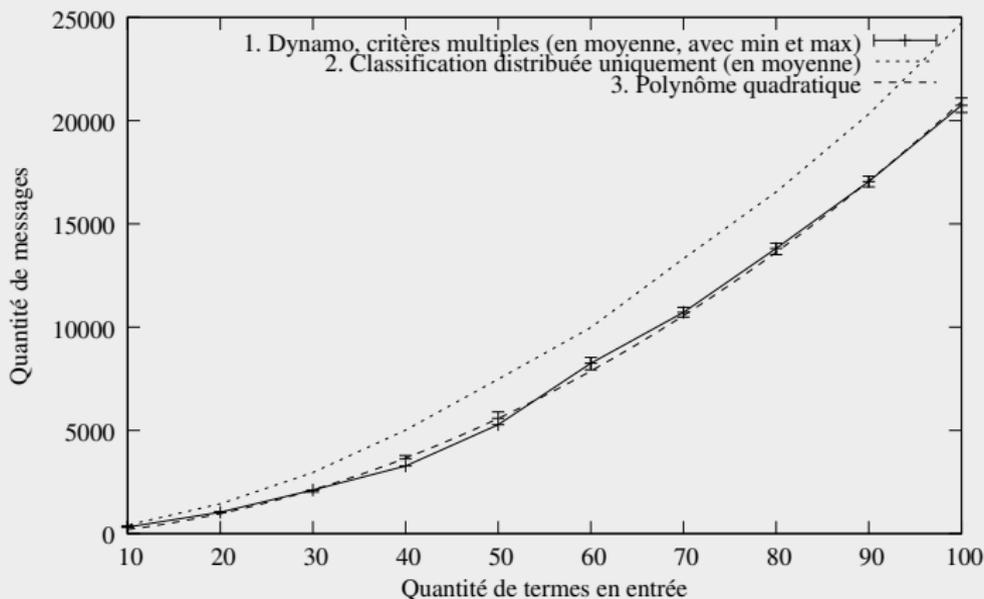
- Rester simple
  - Critères locaux
  - Valeurs nominales pour ces critères
- AMAS : utiliser l'heuristique de la coopération

### Coopération

- Minimiser la non-coopération
- Système à priorités
  - Déterminer tous les problèmes en cours
  - Trouver le plus critique
  - Essayer de le résoudre

# Gérer plusieurs critères

## Impact sur la complexité



- Complexité en moyenne:  $O(n^2)$
- Écart-type maximum: environ 0,6%

# Passage aux arbres n-aires

## Comportement recherché

### Observations

- La classification pousse à un arbre binaire
- Le profil de l'arbre est conditionné par la stratégie de vote des agents

### But

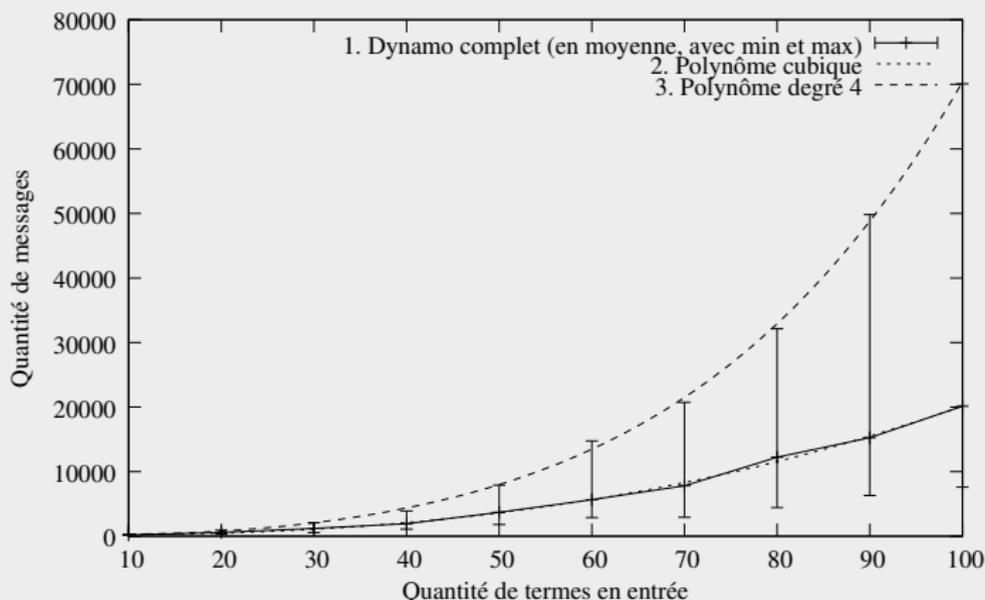
- Permettre d'obtenir des arbres n-aires

### Nouvelle stratégie de vote

- Introduction d'une tolérance  $\varepsilon$  sur la similitude
- Préférence partielle lors des votes en fonction de  $\varepsilon$

# Passage aux arbres n-aires

## Impact sur la complexité



- Complexité en moyenne:  $O(n^3)$
- Complexité au pire cas:  $O(n^4)$

# Plan

- 1 Contexte
- 2 Contributions
- 3 Application
- 4 Perspectives
- 5 Conclusion



# Protocole expérimental

## Données

- Résumés d'articles publiés dans *Astronomy & Astrophysics*
- Ontologie de « référence » fournie

## Difficultés

- Pas de benchmark pour la construction d'ontologies
- Relativité de l'ontologie de « référence »

## Démarche

- Termes en entrée basés sur l'ontologie de « référence »
- Première exécution automatique
- Interventions de l'ontologue pour un résultat « satisfaisant »
- Mesure de *taxonomy overlapping* pour se comparer à l'ontologie de « référence »



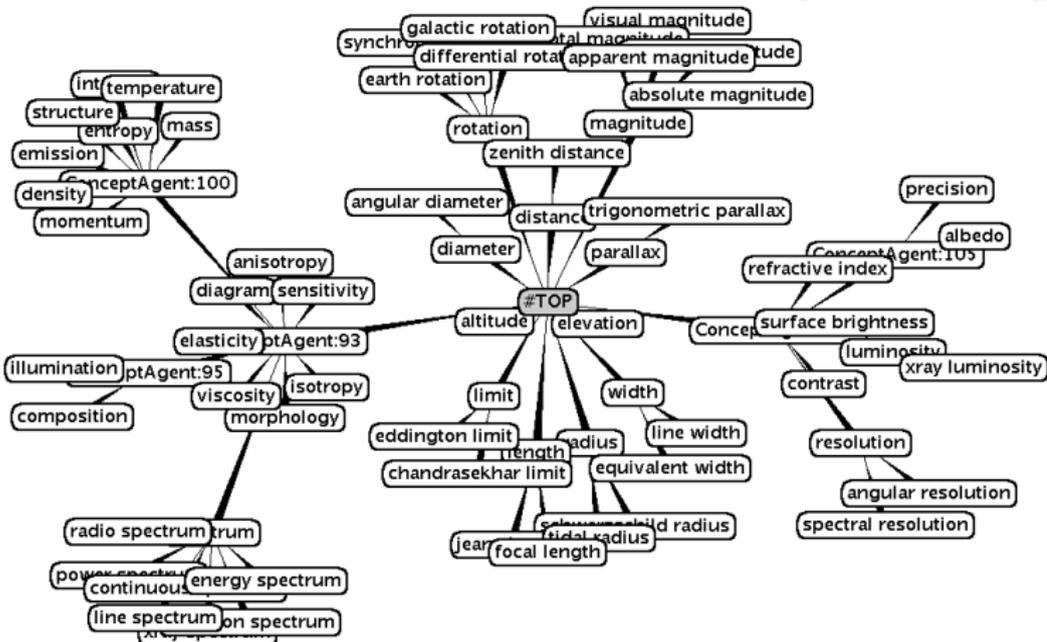


# Déroulement

Après quatre interventions de l'ontologue

## Description

- A la fois du déplacement de feuilles et de branches

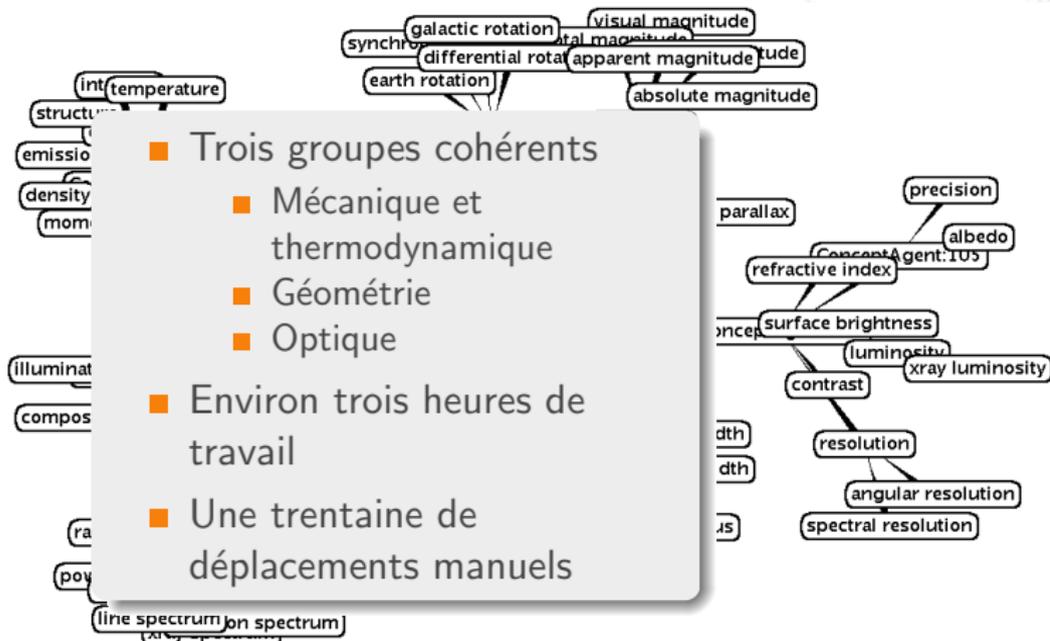


# Déroulement

Après quatre interventions de l'ontologue

## Description

- A la fois du déplacement de feuilles et de branches



# Bilan de l'approche

## Qualités constatées

- Effort réduit de la part de l'ontologue
- Avantages sur la classification « classique »
  - Reprise du système sur les interventions de l'ontologue
  - Prise en compte de critères supplémentaires

## Non validé

- Le résultat produit est-il « meilleur » ?
- Utilisable par un opérateur humain ?

→ besoin de mise à l'épreuve par des ontologues

# Bilan de l'approche

## Qualités constatées

- Effort réduit de la part de l'ontologue
- Avantages sur la classification « classique »
  - Reprise du système sur les interventions de l'ontologue
  - Prise en compte de critères supplémentaires

## Non validé

- Le résultat produit est-il « meilleur » ?
- Utilisable par un opérateur humain ?

→ besoin de mise à l'épreuve par des ontologues

# Bilan de l'approche

## Qualités constatées

- Effort réduit de la part de l'ontologue
- Avantages sur la classification « classique »
  - Reprise du système sur les interventions de l'ontologue
  - Prise en compte de critères supplémentaires

## Non validé

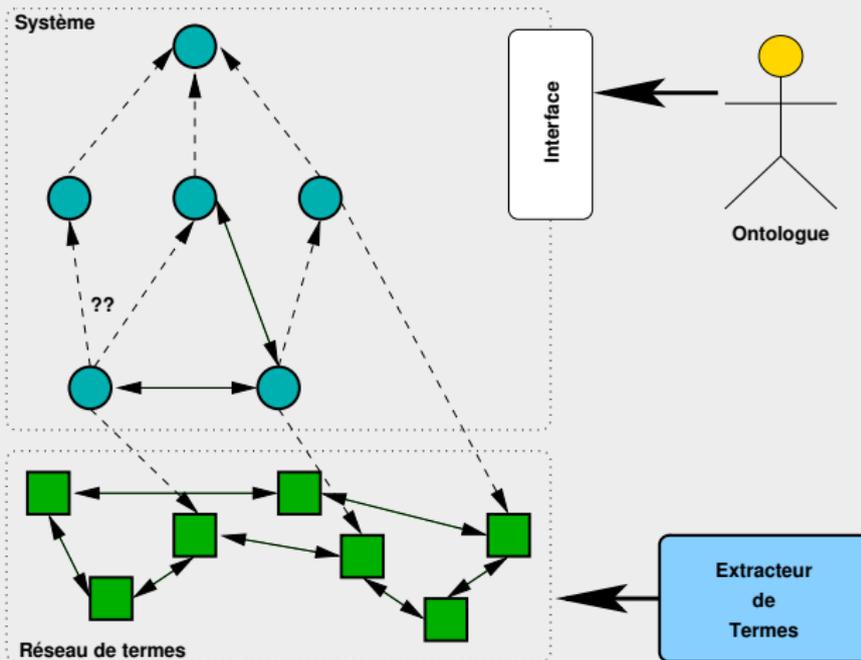
- Le résultat produit est-il « meilleur » ?
  - Utilisable par un opérateur humain ?
- besoin de mise à l'épreuve par des ontologues

# Plan

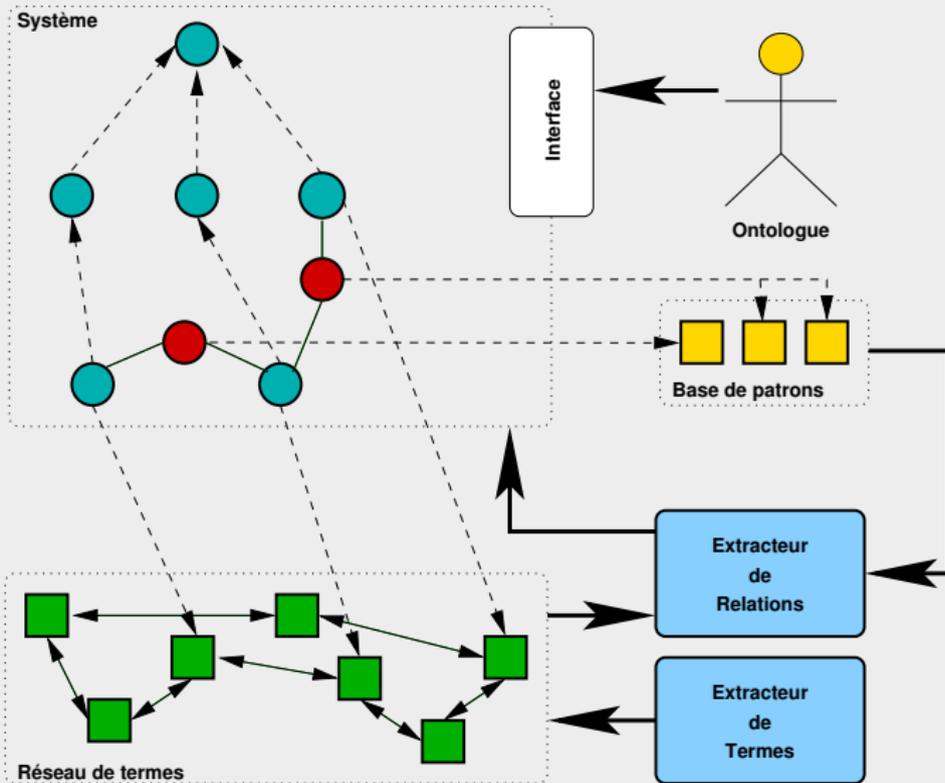
- 1 Contexte
- 2 Contributions
- 3 Application
- 4 Perspectives
- 5 Conclusion



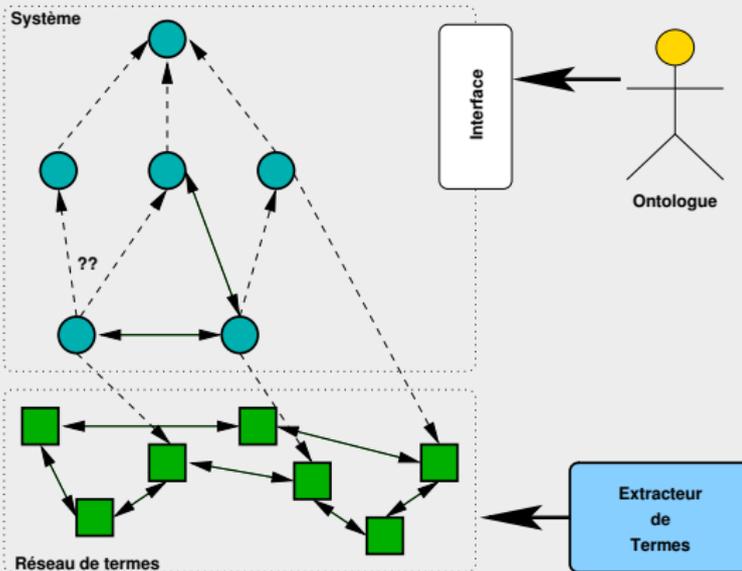
# Relations transverses



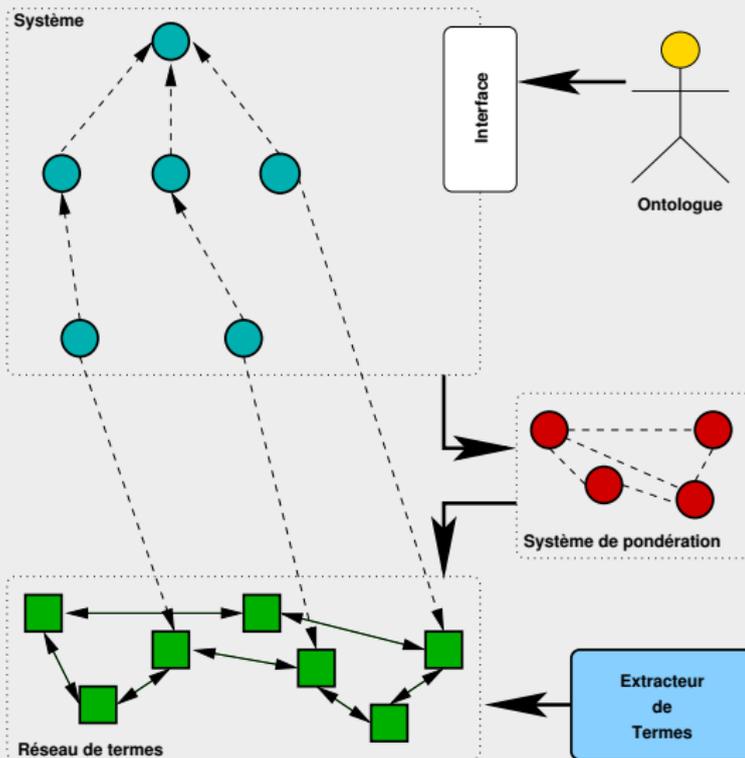
# Relations transverses



# Apprendre sur les règles de calcul



# Apprendre sur les règles de calcul



# Ontologies formelles & Réutilisation

## Problèmes

- Structure taxonomique faiblement formalisée
- Plusieurs solutions structurellement correctes

→ Comment guider l'ontologue?

## Apports de connaissances supplémentaires

- Réutiliser une portion d'ontologie déjà construite
  - DOLCE
  - core-ontologies de domaine
- Permettre d'exprimer des contraintes plus complexes

# Améliorer l'algorithme de classification

## Axes d'amélioration

- Complexité de l'algorithme
  - Stratégie de prédiction des votes ?
- Eliminer la dépendance à l'ordre d'insertion
- Micro-optimisation sur l'implémentation ?

## Curiosité intellectuelle

- Quid d'une implémentation centralisée ?
- Utilisation brute pour de la recherche d'informations ?

# Améliorer l'algorithme de classification

## Axes d'amélioration

- Complexité de l'algorithme
  - Stratégie de prédiction des votes ?
- Eliminer la dépendance à l'ordre d'insertion
- Micro-optimisation sur l'implémentation ?

## Curiosité intellectuelle

- Quid d'une implémentation centralisée ?
- Utilisation brute pour de la recherche d'informations ?

# Enrichir l'interaction avec l'ontologue

## Ergonomie et interaction enrichie

- Comment manipuler un graphe important qui change dans le temps ?
- Comment exprimer des contraintes sur la structuration de l'ontologie ?

## Mise à l'épreuve réelle

- Tester avec des ontologues
- Evaluer l'effort pour passer à une ontologie formelle

# Enrichir l'interaction avec l'ontologue

## Ergonomie et interaction enrichie

- Comment manipuler un graphe important qui change dans le temps ?
- Comment exprimer des contraintes sur la structuration de l'ontologie ?

## Mise à l'épreuve réelle

- Tester avec des ontologues
- Evaluer l'effort pour passer à une ontologie formelle

# Plan

- 1 Contexte
- 2 Contributions
- 3 Application
- 4 Perspectives
- 5 Conclusion



# Les premiers pas sont faits...

## Apports à l'ingénierie des connaissances

- Rénovation de la classification
  - Heuristiques supplémentaires pour les feuilles
  - Approche distribuée
- Première utilisation des SMA dans un tel contexte
- Premiers pas vers la *conception vivante* d'ontologies

## Apports aux AMAS

- Confrontation de la technique à un nouveau domaine
  - Pas d'approche antropomorphique
  - Domaine complexe sans solution connue
- Ingénierie des modèles dans ADELFE
  - Rationalisation du modèle agent
  - Génération de code

# ... mais le chemin reste à parcourir

## Plus de validation

- Produire une interface utilisable
- Mettre à l'épreuve auprès d'ontologues et sur différents projets

## Approche à compléter

- Extraire les relations transverses
- Intégrer dans une chaîne de production d'ontologies
- Dynamo : projet ANR

# Questions ?

Kévin Ottens  
[ottens@irit.fr](mailto:ottens@irit.fr)

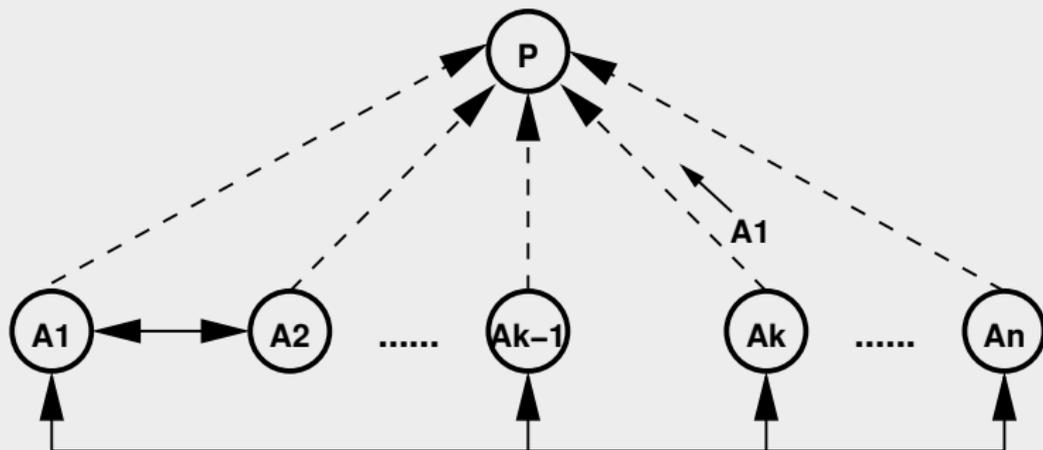
# Questions ?

Kévin Ottens  
[ottens@irit.fr](mailto:ottens@irit.fr)



# Algorithme distribué de classification [Ottens 07]

## Vue locale



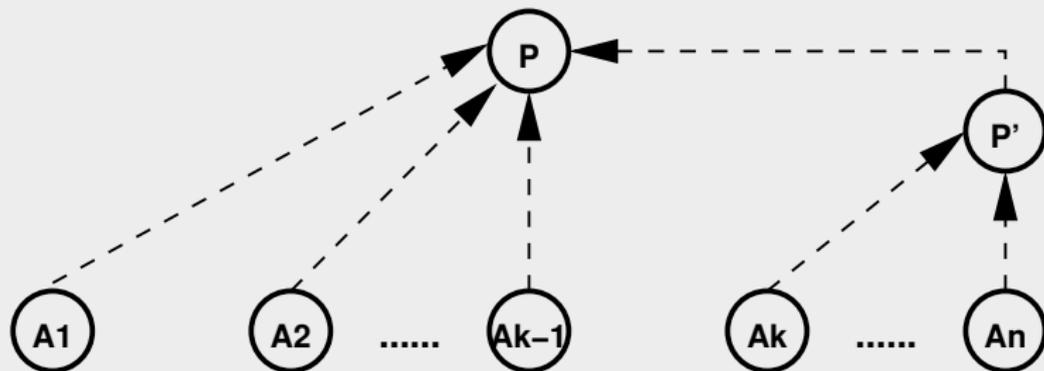
## Etapes

- 1 Evaluation des similarités et « votes »
- 2 Partitionnement et création du niveau intermédiaire
- 3 Changement de parent



# Algorithme distribué de classification [Ottens 07]

## Vue locale

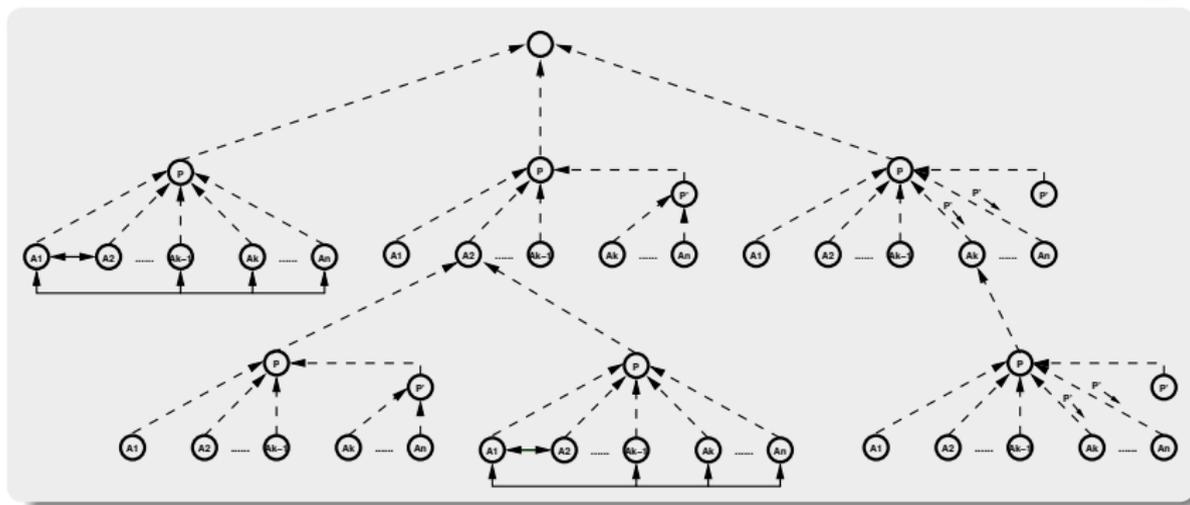


## Etapes

- 1 Evaluation des similarités et « votes »
- 2 Partitionnement et création du niveau intermédiaire
- 3 **Changement de parent**

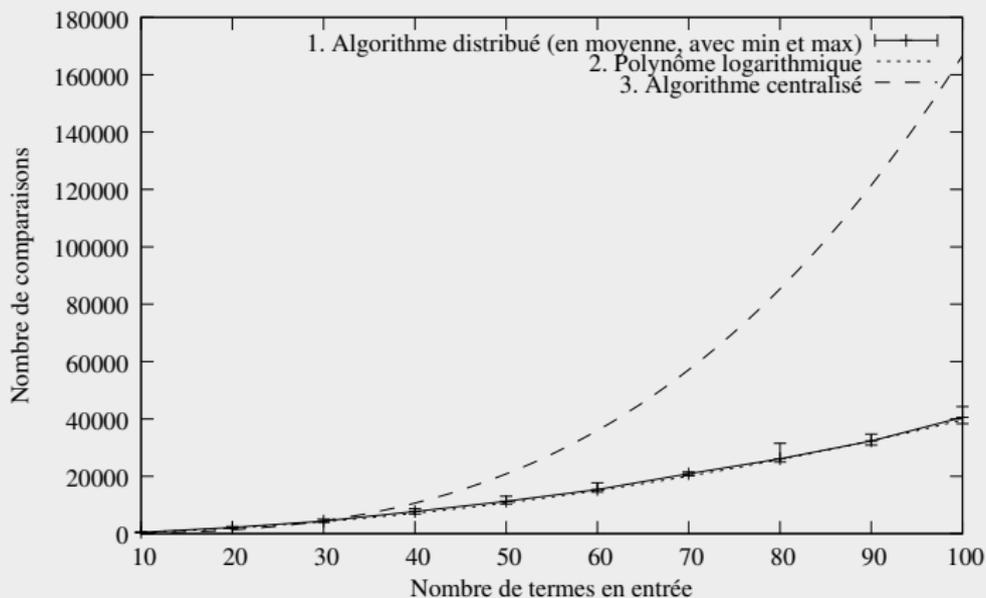
# Algorithme distribué de classification

## Vue globale



# Algorithme distribué de classification

## Complexité expérimentale



- Complexité en moyenne:  $O(n^2 \log(n))$
- Ecart-type maximum: environ 5%

# Algorithme distribué de classification

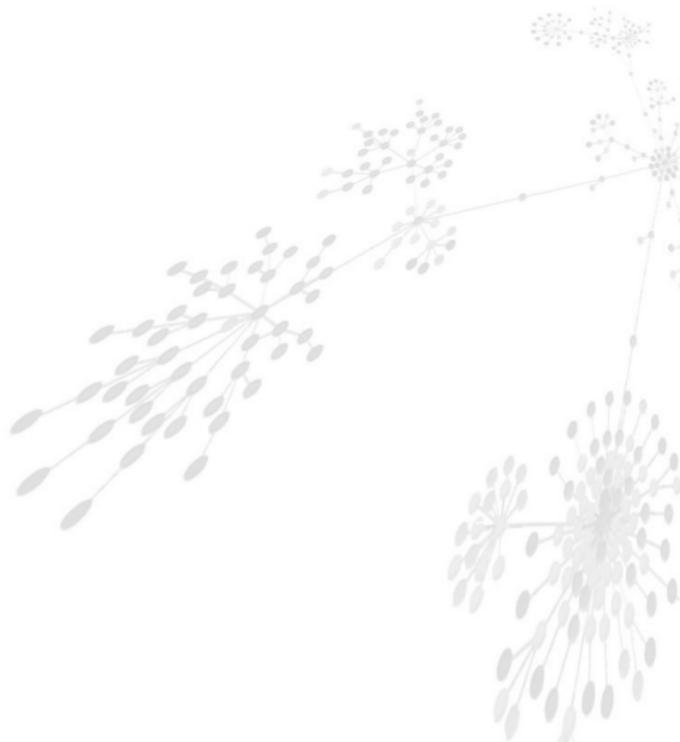
## Point de vue qualitatif

### Exécution automatique

- Vue permanente de la hiérarchie en construction
- Permet d'obtenir un « premier brouillon »

### Intervention de l'ontologue

- Pas d'ajustement de l'algorithme requis
- Dynamicité, la structure est révisée



# Règle de couverture en tête [Ottens 07]

## Comportement recherché

### Observations

- Il existe des paires de termes où la similitude n'est pas définie
- Les ontologues ont des heuristiques spécifiques à la structuration des feuilles

### Buts

- Pouvoir placer ces termes
- Implémenter une heuristique similaire

### Fonction d'adéquation du parent

- Le meilleur parent pour  $E$  est l'agent  $P$  qui maximise  $a(P, E)$
- *Quand un agent  $E$  est insatisfait de son père  $P$ , il évalue  $a(F_i, E)$  avec tous ses frères (notés  $F_i$ ). Celui maximisant  $a(F_i, E)$  est alors choisi comme nouveau parent*

# Règle de couverture en tête

## Détails

### Notations

- $T_X$ : ensemble des termes décrivant l'agent concept  $X$
- $tete(T_X)$ : ensemble de tous les termes qui sont en tête d'au moins un des éléments de  $T_X$

### Fonction d'adéquation d'un parent

- $a(P, E) = \frac{|T_P \cap tete(T_E)|}{|T_P \cup tete(T_E)|}$
- Le meilleur parent pour  $E$  est l'agent  $P$  qui maximise  $a(P, E)$ 
  - plus  $a(P, E)$  est grand plus le recouvrement entre les termes de  $P$  ( $T_P$ ) et les termes tête de  $E$  ( $tete(T_E)$ ) est important
  - or, le père d'un concept est adéquat lorsque l'ensemble de ses termes est un sur-ensemble de l'ensemble des termes têtes de son enfant
  - la part des termes non partagés devant être minimale



# Gérer plusieurs critères

## Guide

### Comment?

- Rester simple
  - Critères locaux
  - Valeurs nominales pour ces critères
- Utiliser l'heuristique de la coopération

### Coopération

- Minimiser la non-coopération
- Système à priorités
  - Déterminer tous les problèmes en cours
  - Trouver le plus critique
  - Essayer de le résoudre

# Gérer plusieurs critères

## Implémentation

### Minimiser la non-coopération

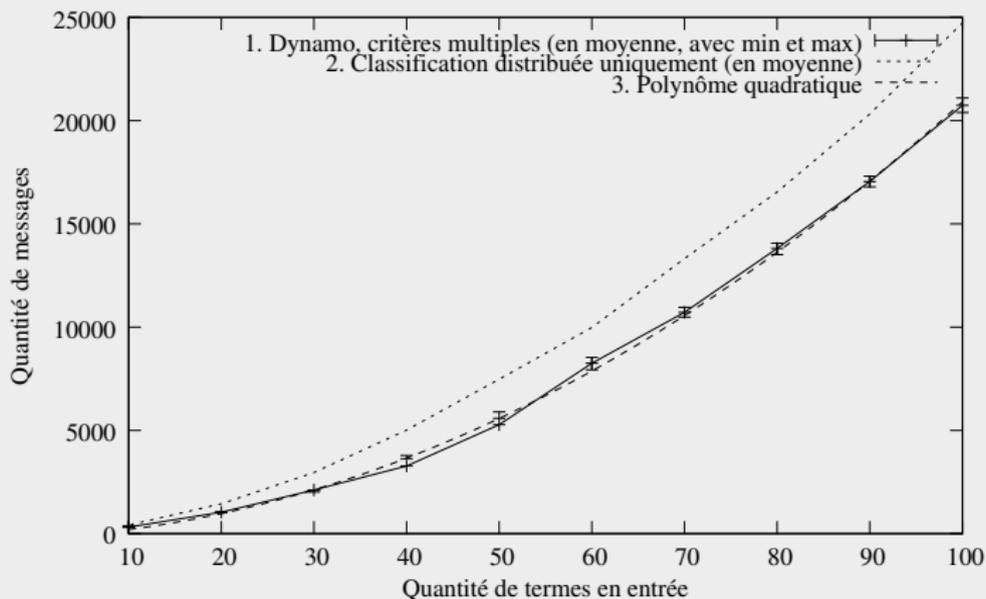
- $\mu_T(A)$ : niveau de non-coopération de « couverture en tête » pour  $A$
- $\mu_F(A)$ : niveau de non-coopération de la « fratrie » pour  $A$
- $\mu_M(A)$ : niveau de non-coopération par « message » pour  $A$
- $\mu(A) = \max(\mu_T(A), \mu_F(A), \mu_M(A))$

### Prendre en charge le problème le plus critique

- $\mu(A) = \mu_T(A) \rightarrow$  Essayer de trouver un meilleur parent
- $\mu(A) = \mu_F(A) \rightarrow$  Améliorer la structure par la classification
- $\mu(A) = \mu_M(A) \rightarrow$  Traiter le message

# Gérer plusieurs critères

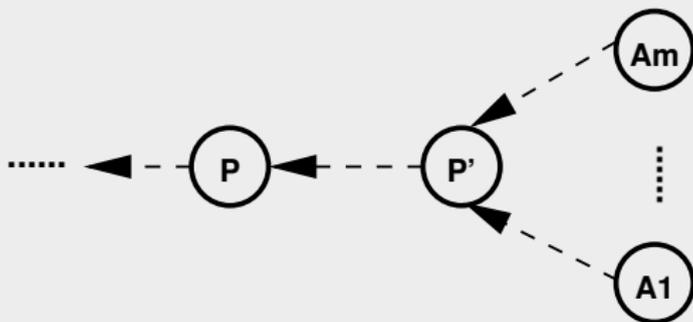
## Impact sur la complexité



- Complexité en moyenne:  $O(n^2)$
- Ecart-type maximum: environ 0,6%



# Nettoyage de l'arborescence

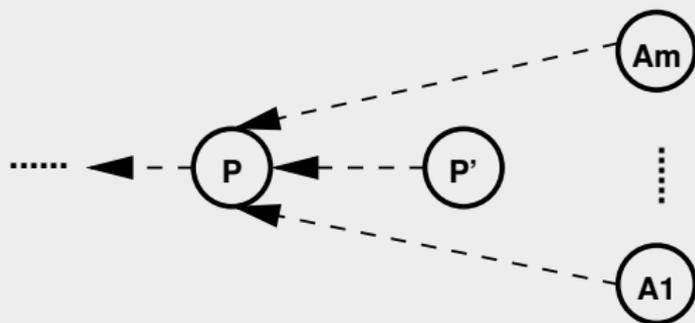


## Etapes

- 1  $P'$  n'a aucun frère, il propose à ses fils de remonter
- 2  $P'$  n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

■ Complexité :  $O(1)$

# Nettoyage de l'arborescence

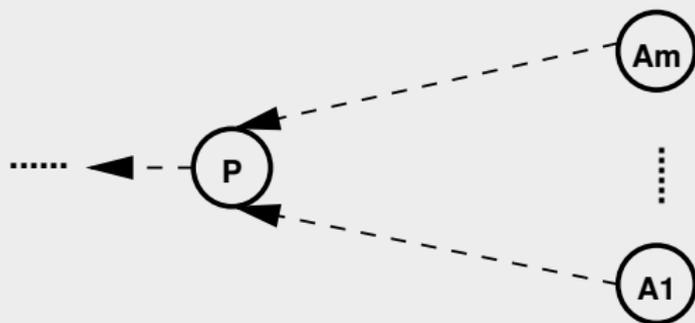


## Etapes

- 1  $P'$  n'a aucun frère, il propose à ses fils de remonter
- 2  $P'$  n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

■ Complexité :  $O(1)$

# Nettoyage de l'arborescence

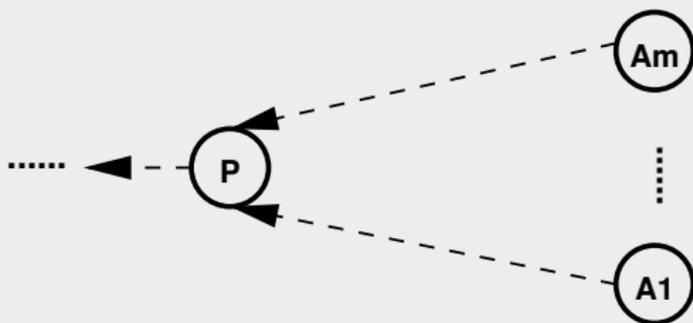


## Etapas

- 1  $P'$  n'a aucun frère, il propose à ses fils de remonter
- 2  $P'$  n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

■ Complexité :  $O(1)$

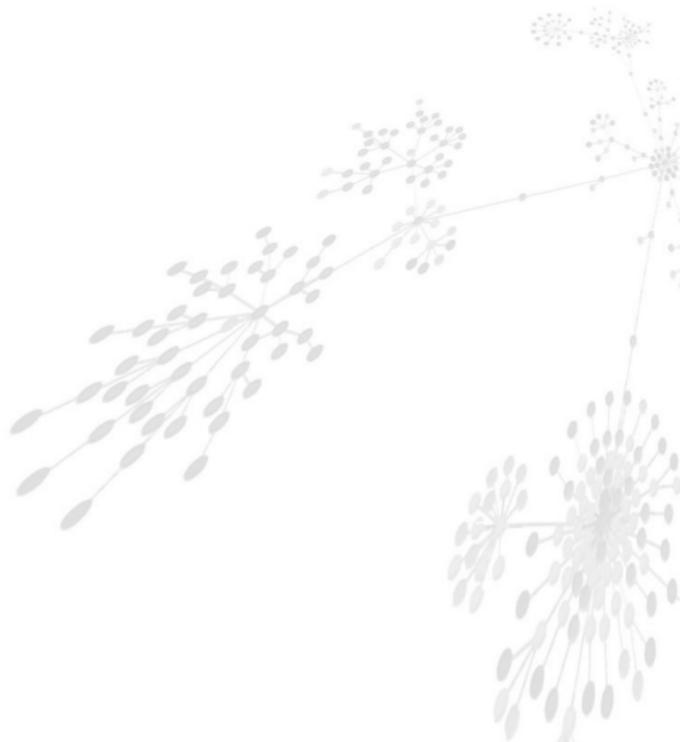
# Nettoyage de l'arborescence



## Etapas

- 1  $P'$  n'a aucun frère, il propose à ses fils de remonter
- 2  $P'$  n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

■ Complexité :  $O(1)$



# Passage aux arbres n-aires

## Comportement recherché

### Observations

- La classification pousse à un arbre binaire
- Le profil de l'arbre est conditionné par la stratégie de vote des agents

### But

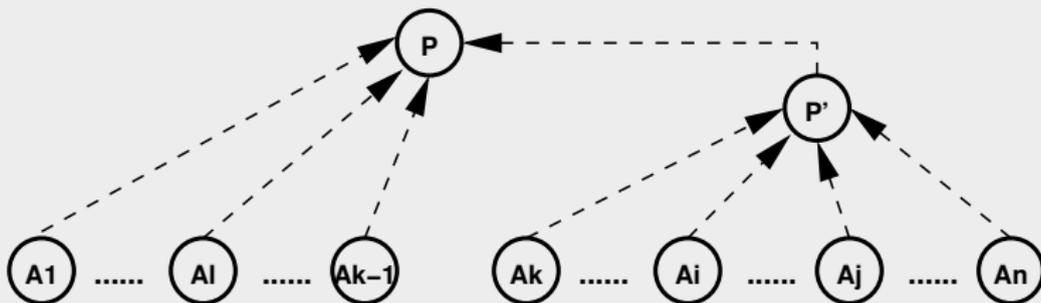
- Permettre d'obtenir des arbres n-aires

### Nouvelle stratégie de vote

- Introduction d'une tolérance  $\varepsilon$  sur la similitude
- Préférence partielle lors des votes en fonction de  $\varepsilon$

# Passage aux arbres n-aires

## Détails

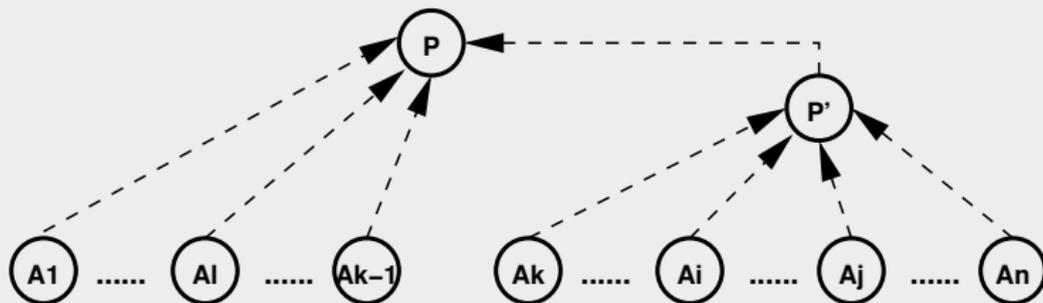


## Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l)$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils  $\rightarrow$  baisser  $\epsilon_P$
  - Trop peu de fils  $\rightarrow$  augmenter  $\epsilon_P$

# Passage aux arbres n-aires

## Détails

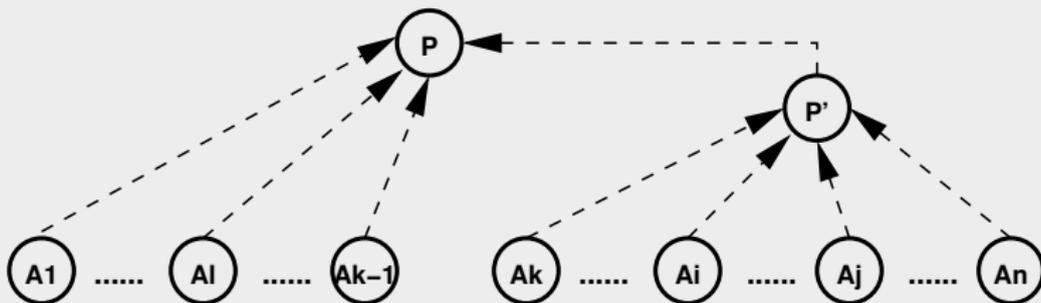


## Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils  $\rightarrow$  baisser  $\varepsilon_P$
  - Trop peu de fils  $\rightarrow$  augmenter  $\varepsilon_P$

# Passage aux arbres n-aires

## Détails

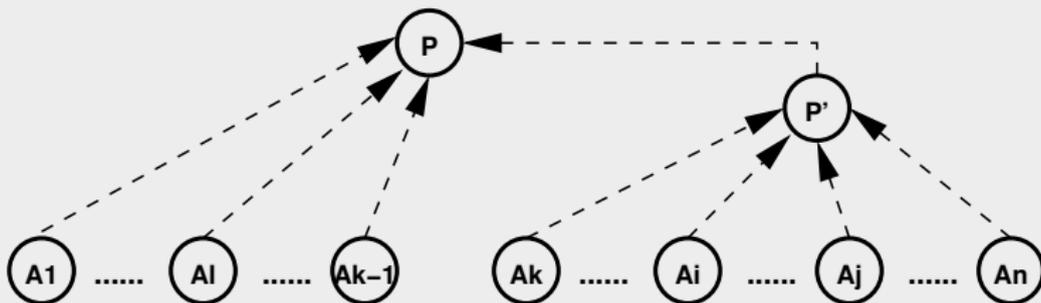


## Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon_{P'}$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils  $\rightarrow$  baisser  $\varepsilon_{P'}$
  - Trop peu de fils  $\rightarrow$  augmenter  $\varepsilon_{P'}$

# Passage aux arbres n-aires

## Détails

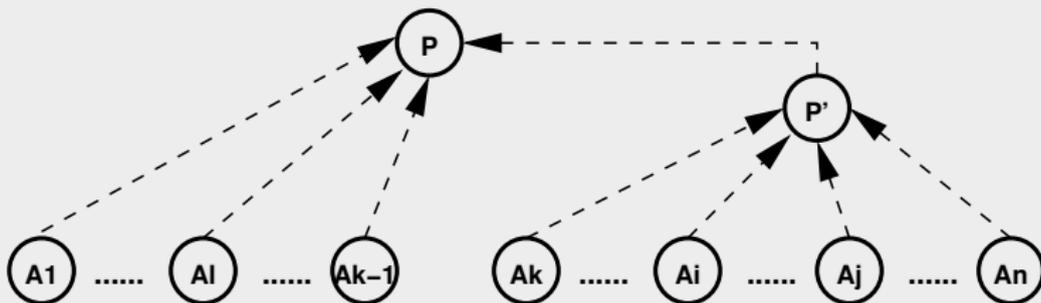


## Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon_{P'}$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils  $\rightarrow$  baisser  $\varepsilon_{P'}$
  - Trop peu de fils  $\rightarrow$  augmenter  $\varepsilon_{P'}$

# Passage aux arbres n-aires

## Détails

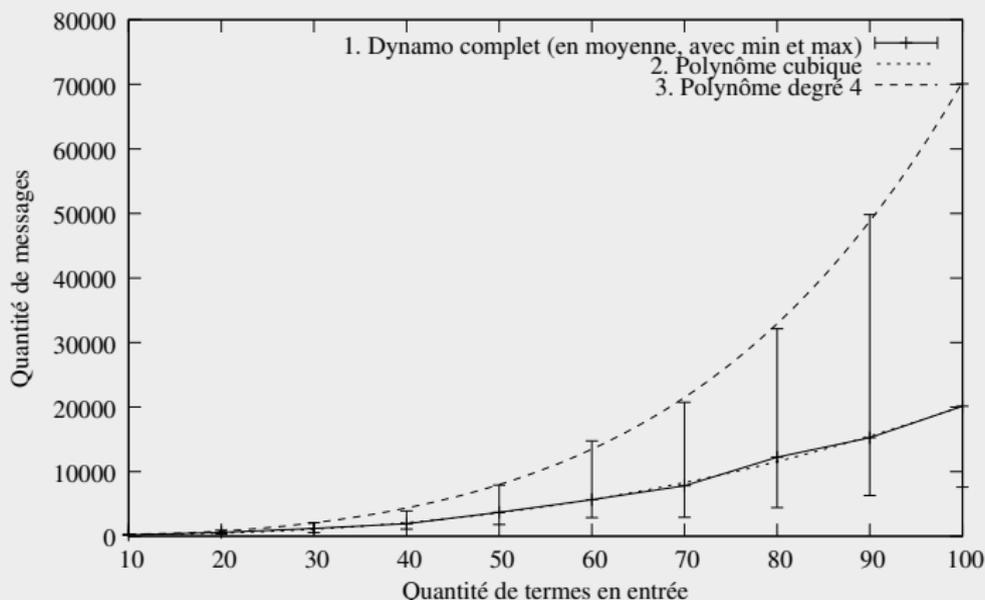


## Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon_{P'}$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils  $\rightarrow$  baisser  $\varepsilon_{P'}$
  - Trop peu de fils  $\rightarrow$  augmenter  $\varepsilon_{P'}$

# Passage aux arbres n-aires

## Impact sur la complexité



- Complexité en moyenne:  $O(n^3)$
- Complexité au pire cas:  $O(n^4)$



# Déroulement

Exécution automatique

## Description

- 90 termes en entrée
- taxonomy overlapping* : 0,78
- placement dans les couches basses comparable à la « référence »



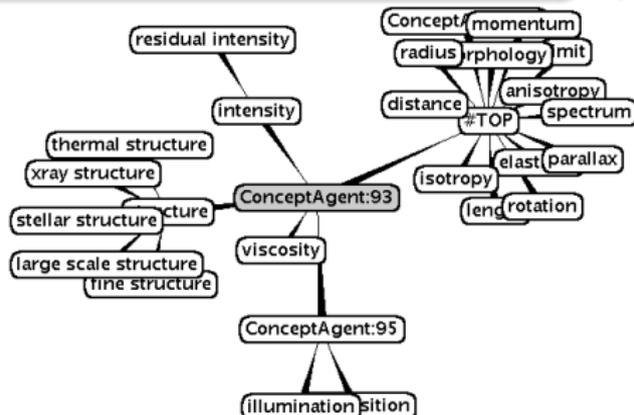


# Déroulement

## Première intervention de l'ontologue

### Description

- Les sous-branches sont ramenées à la racine
- Le système se stabilise dans une configuration « plus plate »
- *taxonomy overlapping* : 0,80
- « ConceptAgent:93 » concentre des propriétés mécaniques et thermodynamiques



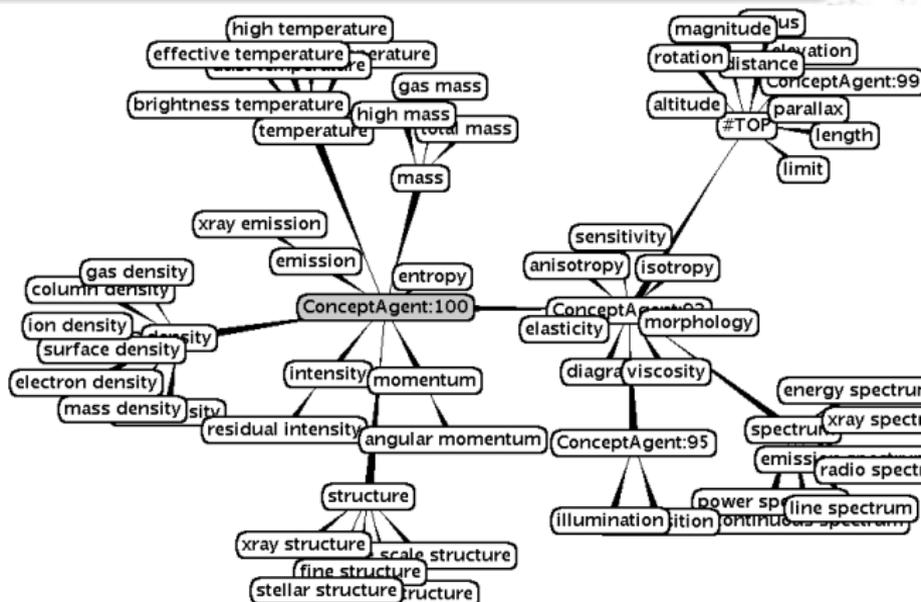
# Déroulement

## Deuxième intervention de l'ontologue

### Description

- Les propriétés mécaniques et thermodynamiques égarées sont ramenées sur « ConceptAgent:93 »

- taxonomy overlapping* : 0,76
- Stabilisation d'une branche plus raffinée



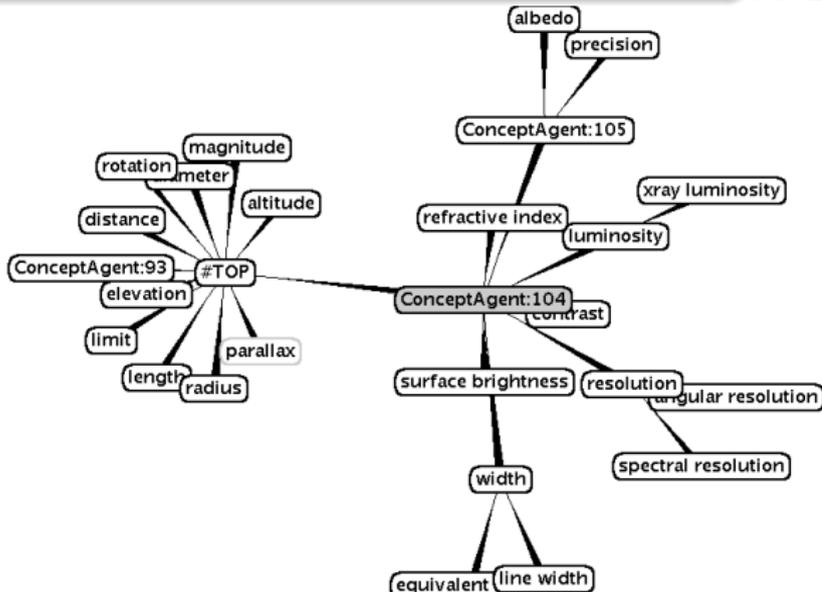
# Déroulement

Troisième intervention de l'ontologue

## Description

- Tri des propriétés géométriques vers la branche concernée
- Sous-branche des propriétés optiques ramenée sous la racine

- Stabilisation du système avec les propriétés géométriques à la racine
- Branche des propriétés optiques

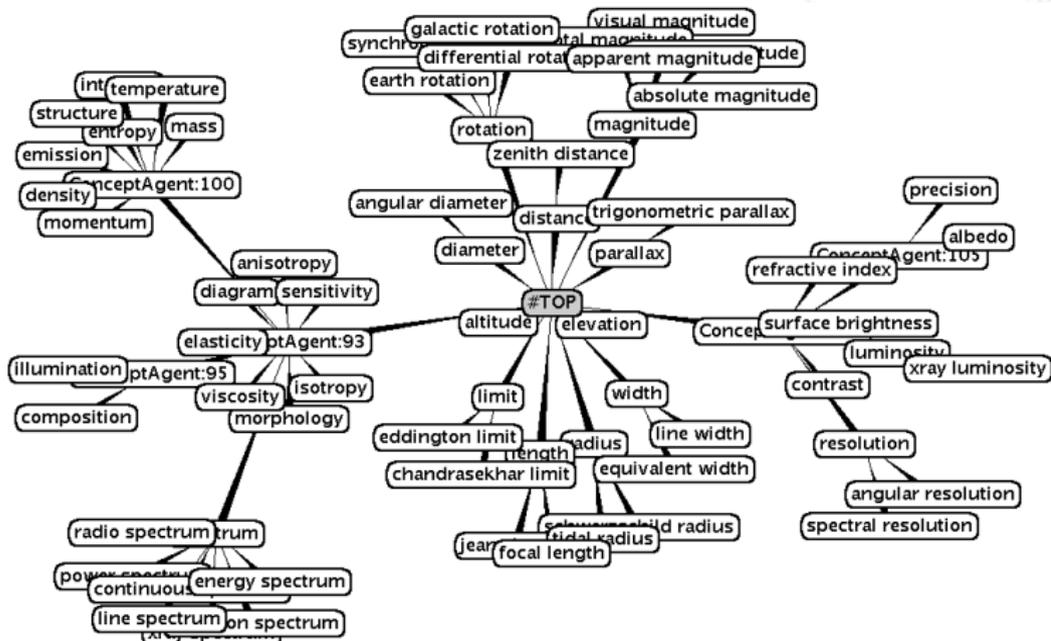


# Déroulement

Quatrième intervention de l'ontologue

## Description

- Déplacement de *width* à la racine

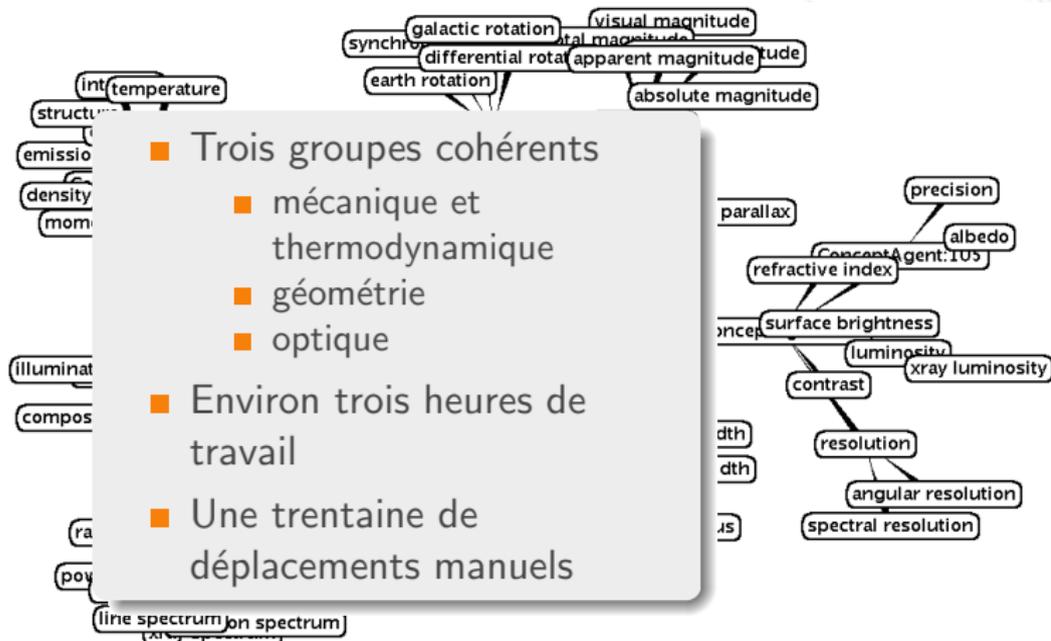


# Déroulement

Quatrième intervention de l'ontologue

## Description

- Déplacement de *width* à la racine





## Principe

- Algorithme de classification
- Exploration du réseau Tête-Expansion
  - Détermine des contextes partagés
  - Permet le calcul d'un indice de similarité

## Principe

- Algorithme de classification
- Exploration du réseau Tête-Expansion
  - Détermine des contextes partagés
  - Permet le calcul d'un indice de similarité

|           | ADMISSIBLE | NOMINAL | HT | THT | 63KV |
|-----------|------------|---------|----|-----|------|
| CÂBLE     | 0          | 0       | 1  | 0   | 1    |
| INTENSITÉ | 1          | 1       | 0  | 0   | 0    |
| LIGNE     | 0          | 0       | 1  | 1   | 1    |
| TENSION   | 1          | 1       | 0  | 0   | 0    |

## Principe

- Algorithme de classification
- Exploration du réseau Tête-Expansion
  - Détermine des contextes partagés
  - Permet le calcul d'un indice de similarité

## Indice de similarité entre $t1$ et $t2$

- $a$  : nombre de contextes partagés par  $t1$  et  $t2$
- $b$  : nombre de contextes présents uniquement chez  $t1$
- $c$  : nombre de contextes présents uniquement chez  $t2$
- $d$  : nombre de contextes présents ni chez  $t1$ , ni chez  $t2$

$$\text{sim}(t1, t2) = \frac{\alpha}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) + \frac{1-\alpha}{2} \left( \frac{d}{d+c} + \frac{d}{d+b} \right)$$

## Principe

- Algorithme de classification
- Exploration du réseau Tête-Expansion
  - Détermine des contextes partagés
  - Permet le calcul d'un indice de similarité

## Indice de similarité entre $t1$ et $t2$

- $a$  : nombre de contextes partagés par  $t1$  et  $t2$
- $b$  : nombre de contextes présents uniquement chez  $t1$
- $c$  : nombre de contextes présents uniquement chez  $t2$
- $d$  : nombre de contextes présents ni chez  $t1$ , ni chez  $t2$

$$\text{sim}(t1, t2) = \frac{\alpha}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) + \frac{1-\alpha}{2} \left( \frac{d}{d+c} + \frac{d}{d+b} \right)$$