



## Thème: **Hybridation de l'Answer Set Programming et de la théorie de Dempster Shafer**

Auteurs : Serge SONFACK S. Laurent Geneste, Bernard Kamsu Foguem

# Plan

- 1 Introduction
- 2 Représentation et raisonnement
- 3 Incertitude
- 4 Hybridation
- 5 Conclusion

# Problème en entreprise

## Resoudre les problèmes

Les entreprises sont confrontées à des problèmes qui peuvent avoir des répercussions sur leurs activités principales



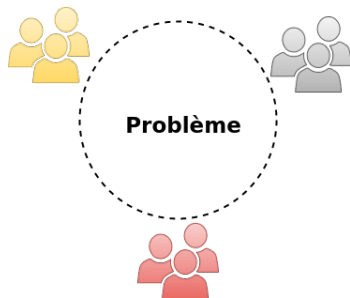
Généralement elles n'ont pas de solution à portée de main et vont interpeller les experts dans le but de :

- Comprendre les causes
- Résoudre le problème

# Expertise

## Expertises

Appel aux experts pour réaliser des **expertises**



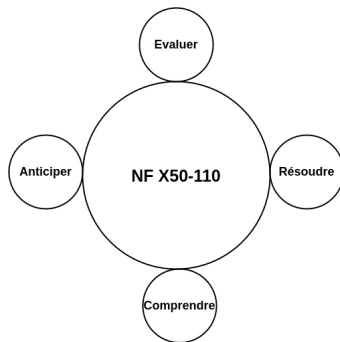
Appliquer leurs connaissances :

- Expliquer
- Résoudre
- Évaluer

# Standardisation

## NF X50-110

Il s'agit d'un ensemble de bonnes pratiques pour les experts [4] :



Définit un processus d'expertise :

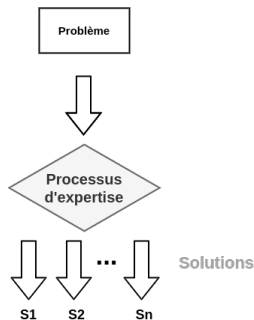
- **Explorer un problème**
- **Faire ressortir les différentes solutions**

## Processus NF X50-110

### Processus exploratoire

Définit un processus d'expertise exploratoire :

- Guidé par les hypothèses
- Explorant toutes les possibilités



Retrouver toutes les solutions possibles : S1, S2, ..., Sn

## Programmation par ensembles solutions : ASP

### ASP [2]

- Langage simple riche et performant :
  - Offre de cardinalités sur les contraintes
- ASP est un paradigme de programmation déclaratif :
  - Règles
  - Contraintes
  - Faits
- Représentation des connaissances et raisonnement
- Utilisé pour résoudre des problèmes tels que la configuration de produits, l'aide à la décision, la composition musicale, la constitution d'équipes, la planification etc

### ASP pour processus d'expertise

**Simple + flexible + problème de recherche combinatoire + solveur rapide**(Clasp, Clingo, DLV, Smodels)

## ASP-Vue globale

## Vue globale

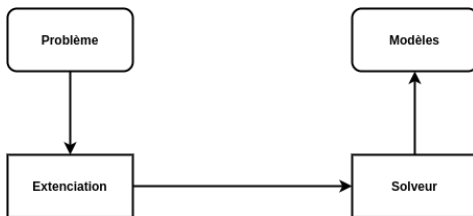


Figure: ASP workflow

- Langage déclaratif
- Représentation de la connaissance
- Utilise un solveur
- Raisonnement **non-monotone**



## ASP-Syntaxe

## Syntaxe

Une règle  $r$  est de la forme :

$$a_0 \leftarrow b_1, \dots, b_m, \text{not}b_{m+1}, \dots, \text{not}b_n \quad (1)$$

$$\text{tete}(r) = \{a_0\}$$

$$\text{corps}(r) = \{b_1, \dots, b_m, \text{not}b_{m+1}, \dots, \text{not}b_n\}$$

- Basé sur la **négation par défaut**  
 $A \leftarrow B, \text{not } C.$   
**not**: *on ne croit pas que*
- Contrainte d'intégrité
- Règle de choix

$$I\{a_1; \dots; a_m; \text{nota}_{m+1}; \dots; \text{nota}_n\} u \leftarrow b_1, \dots, b_m, \text{not}b_{m+1}, \dots, \text{not}b_n \quad (2)$$

# ASP-Sémantique

## Sémantique

La sémantique ASP est basée sur la sémantique *modèle stable*, qui calcule les ensembles de solutions d'un programme, également appelés ses modèles

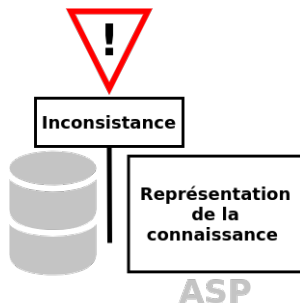
$S$  une instantiation d'atomes d'un programme  $P$

- $S$  est un modèle de  $P$  si :
  - $tete(r) \in S$ , lorsque  $corps^+(r) \subseteq S$  et  $corps^-(r) \cap S = \emptyset \forall r \in P$  [2].
- Une approche permettant de retrouver ces modèles est de passer par :
  - 1 **La réduction Gelfond-Lifschitz**  
 Qui va d'une instantiation de  $P$ , pour sa réduction en programme **défini** (*sans négation*)  
 $P^S = \{tete(r) \leftarrow corps^+(r) / r \in P, corps^-(r) \cap S = \emptyset\}$
  - 2 **Déduction**  
 La déduction d'un programme  $P$  notée  $CI(P)$  est l'ensemble des conséquences déduites des règles de  $P$
  - 3 Si  $S$  coïncide avec  $CI(P)$  alors  $S$  est un modèle de  $P$

# Inconsistance

## Mesure

Inconsistance peut se définir comme la présence de contradictions dans une base de connaissance, par rapport au formalisme de représentation [1]



Les causes :

- Manque d'information
- Données erronées

## Mesure de l'inconsistance

### Mesure

Fonction qui permet de connaître le niveau de sévérité de l'inconsistance dans une connaissance :

- Fonction de mesure :

$$I_{\pm}(P) = \min\{|A| + |D|\} \quad [6]$$

$P$  : programme

$A$  : règles ajoutées de  $P$ ,  $D$  : règles réduites de  $P$

tel que  $(P \cup A) - D$  est consistant

- Complexité : NP

### Mesure drastique

La mesure drastique [5]

$$I_d(K) = \begin{cases} 1 & \rightarrow \textit{inconsistance} \\ 0 & \end{cases}$$

## Dempster Shafer

## Fondamentaux du DST

Soit  $\Theta = \{\theta_1, \dots, \theta_n\}$  : cadre de discernement (FOD)

$$m : 2^\Theta \rightarrow [0, 1]$$

$$A \mapsto m(A)$$

$$\sum \{m(A) / A \subseteq \Theta\} = 1.$$

$m(\emptyset) = 0$  est la forme normalisée (Hypothèse du monde clos)

- La fonction de croyance  $Bel()$  :

$$\begin{cases} Bel(A) = \sum_{B \subseteq A} m(B) \\ m(\emptyset) = 0 \end{cases}$$

$Bel(A)$  est interprété comme le degré de croyance que la vérité réside en  $A$

- La fonction de plausibilité  $Pl()$  d'une hypothèse  $A$  est la quantité de croyance non strictement engagée dans le complément de  $\bar{A}$

$$\begin{cases} Pl : 2^\Theta \rightarrow [0, 1] \\ Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(\bar{A}), A \subseteq \Theta \end{cases}$$

## DST-ASP

## Hybridation

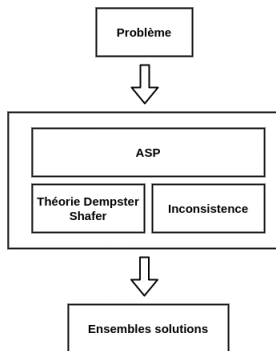


Figure: Approche DST-ASP

## ASP

## Exemple

Soit le problème consistant à rechercher les caractéristiques de d'un être *tweety* :

$oiseau(X) : -vole(X), plume(X), not\ anormal(X).$

$\{vole(X); nage(X)\} : -oiseau(X).$

$oiseau(tweety).$

## Inconsistance

- Ne peut pas *nage* et *vole* simultanément
- :  $-nage(X), vole(X).$

## Herbrand

La base de Herbrand associée à ce programme est:

$B_P = \{vole(tweety), oiseau(tweety), anormal(tweety), plume(tweety), nage(tweety)\}$

## DST

## Distribution

{ vole(tweety) }: 0.1	{ oiseau(tweety)}: 0.2
{ nage(tweety)}: 0.1	{ plume(tweety)}: 0.1
{ vole(tweety) , nage(tweety)}:0.1	{ vole(tweety) , oiseau(tweety)}: 0.2
{ vole(tweety) , plume(tweety)}:0.1	{nage(tweety) , oiseau(tweety)}: 0.1

## Ensemble-Croyance

- 1  $X1 = \{oiseau(tweety)\}$  avec une croyance de :  
 $Bel(X1) = 0.2$
- 2  $X2 = \{oiseau(tweety), vole(tweety)\}$   
 $Bel(X2) = 0.2 + 0.1 + 0.2 = 0.5$
- 3  $X3 = \{oiseau(tweety), nage(tweety)\}$   
 $Bel(X3) = 0.2 + 0.1 + 0.1 = 0.4$



# Décision

## Valeur de décision [3]

- 1  $X1 = \{oiseau(tweety)\}$  La croyance de  $X1$  par rapport à la distribution  $m$  est :

$$Bel(X1) = 0.2$$

$$w_1 = 0.5, w_2 = 0.5$$

$$GOWA_{w,1}(P, X) = 0.5 * 0.25 + 0.5 * 0.2 = 0.22$$

$$I1 = 0.22$$
- 2  $X2 = \{oiseau(tweety), vole(tweety)\}$

La croyance de  $X2$  par rapport à la distribution  $m$  est :

$$Bel(X2) = 0.5$$

$$w_1 = 0.5, w_2 = 0.5$$

$$GOWA_{w,1}(P, X) = 0.5 * 0.25 + 0.5 * 0.5 = 0.37$$

$$I2 = 0.37$$
- 3  $X3 = \{oiseau(tweety), nage(tweety)\}$

La croyance de  $X3$  par rapport à la distribution  $m$  est :

$$Bel(X3) = 0.4$$

$$w_1 = 0.5, w_2 = 0.5$$

$$GOWA_{w,1}(P, X) = 0.5 * 0.25 + 0.5 * 0.4 = 0.32$$

$$I3 = 0.32$$

## Conclusion

### Conclusion

- NF X50 110 : Processus d'expertise
- Représentation de la Connaissance : \*ASP, \*Inconsistance
- Incertitude : DST
- Hybridation : DST-ASP






Méthode	Solution au problème	Choix d'une solution
ASP	oui	non
DST	non	oui
<b>Hybridation</b>	<b>oui</b>	<b>oui</b>

Table: Comparaison ASP, DST et Hybridation

### Difficultés

\*complexité: mesure drastique-raffinement ——— Intégration au solver ASP \*

## References I

-  Mark Burgin and CNJ de Vey Mestdagh.  
Consistent structuring of inconsistent knowledge.  
[Journal of Intelligent Information Systems](#), 45(1):5–28, 2015.
-  Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub.  
Answer set solving in practice.  
[Synthesis lectures on artificial intelligence and machine learning](#), 6(3):1–238, 2012.
-  Fateme Kouchakinezhad and Alexandra Šipošová.  
Ordered weighted averaging operators and their generalizations with applications in decision making.  
[Iranian Journal of Operations Research](#), 8(2):48–57, 2017.
-  Huver Loisel Peyrouy Pineau Tuffery M. Peyrouy, Chanay; Fourniguet.  
Recommandations pour l'application de la norme nf x 50-110:2003.  
Technical report, Association Française de Normalisation, 2011.
-  Matthias Thimm and Johannes P Wallner.  
On the complexity of inconsistency measurement.  
[Artificial Intelligence](#), 275:411–456, 2019.

## References II



Markus Ulbricht, Matthias Thimm, and Gerhard Brewka.

Measuring inconsistency in answer set programs.

[In European Conference on Logics in Artificial Intelligence](#), pages 577–583. Springer, 2016.