

# Nettoyage de données XML : combien ça coûte ?

---

Laure Berti-Équille

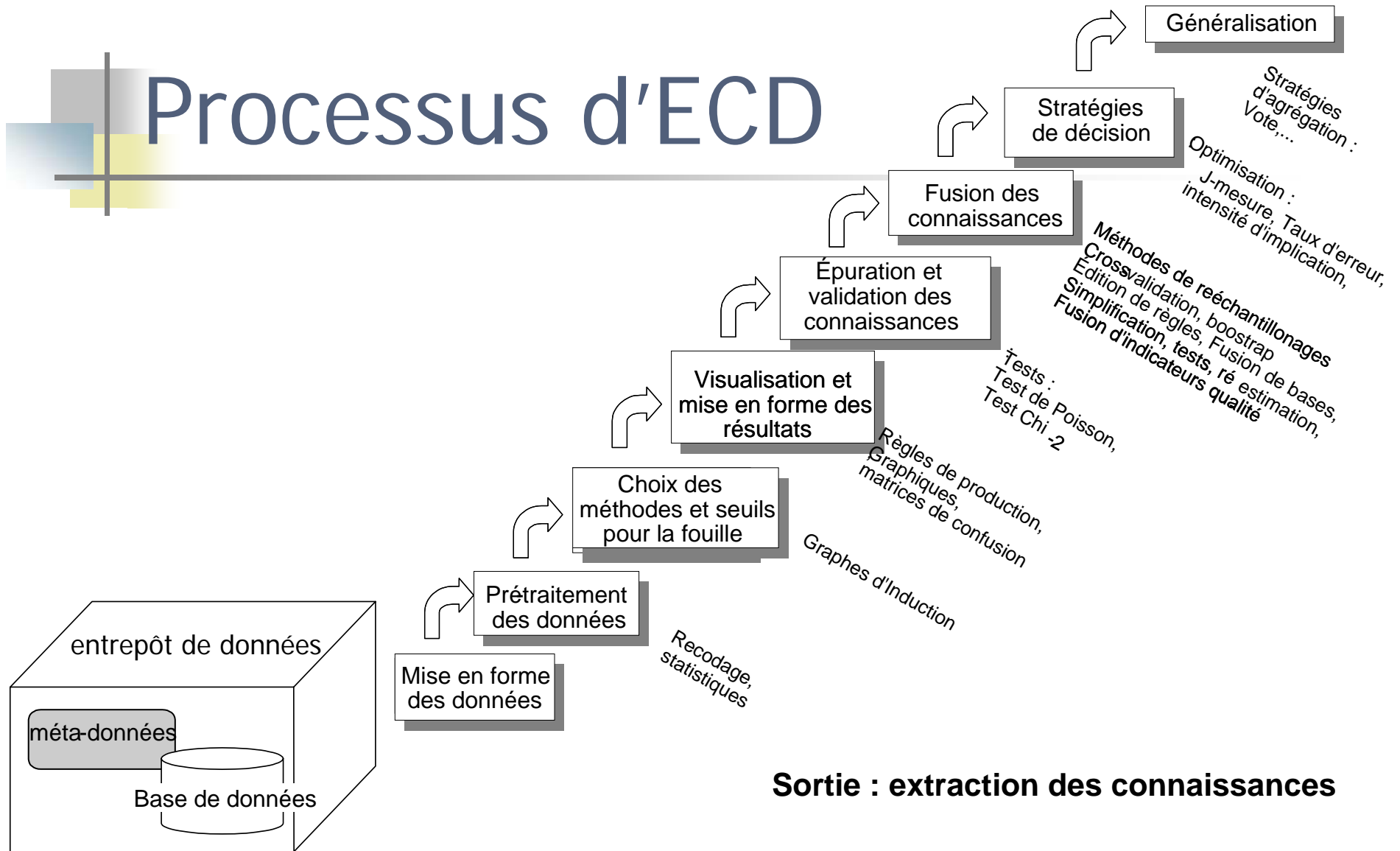
**IRISA, Projet TEXMEX, Campus de Beaulieu**

**35042 Rennes cedex, FRANCE**

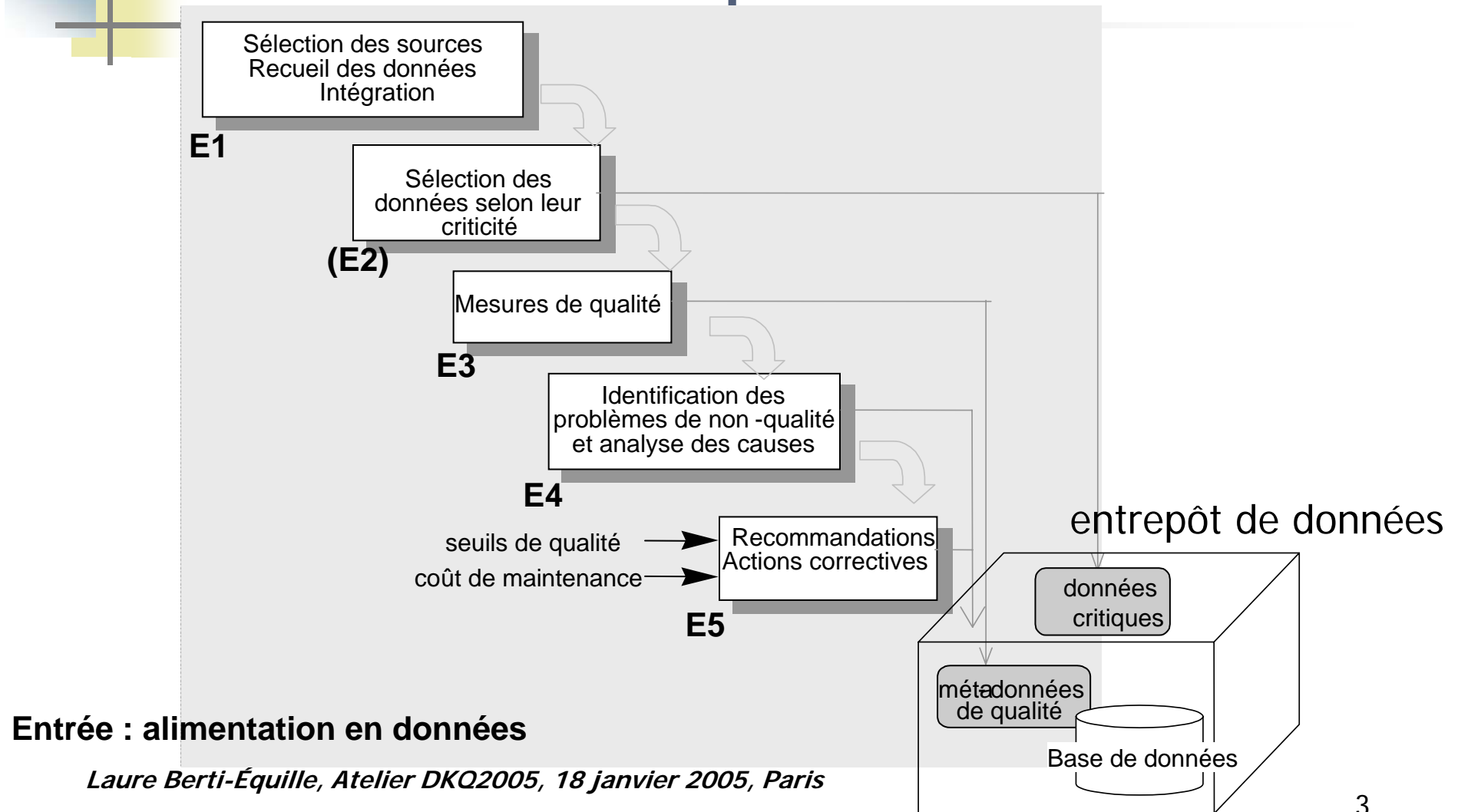
[berti@irisa.fr](mailto:berti@irisa.fr)

<http://ww.irisa.fr/textmex>

# Processus d'ECD



# En amont du processus d'ECD





# Plan

---

- Qualité et nettoyage des données
- Typologie des problèmes de QDD XML
- Définitions préliminaires
- Proposition d'un modèle de coût
- Conclusion et perspectives



# Qualité des données (QDD)

1/2

## ■ Enjeux scientifiques, industriels et financiers

- Certification des systèmes
- Échange des données et standardisation de méta-données
- Qualité des entrepôts de données et assurance des décisions

## ■ Principales causes de non-qualité de données

- Conception et développement
- Migration et conversion
- Intégration de systèmes hétérogènes
- Erreurs de saisie
- Fraude
- Vieillesse des données
- **Non-satisfaction des utilisateurs**

# Qualité des données (QDD)

2/2

## Principales activités

- ✓ Database Profiling (propriétés statistiques, inférence sur la structure)
- ✓ Record Linkage / Record matching / Object identification
- ✓ Intégration de données (sélection de sources, appariement de schémas, résolution de conflits d'instances, multi-requêtage, fusion de résultats, agrégation de mesures de qualité)
- ✓ Détection d'erreurs, audit des données (contraintes d'intégrité, déviation)
- ✓ Correction des données, nettoyage de données (data cleaning, data scrubbing) ou nettoyage de schéma (activités ETL)
- ✓ Optimisation (par trade-off ou par modèle de coût)

## ■ Paradigmes et techniques

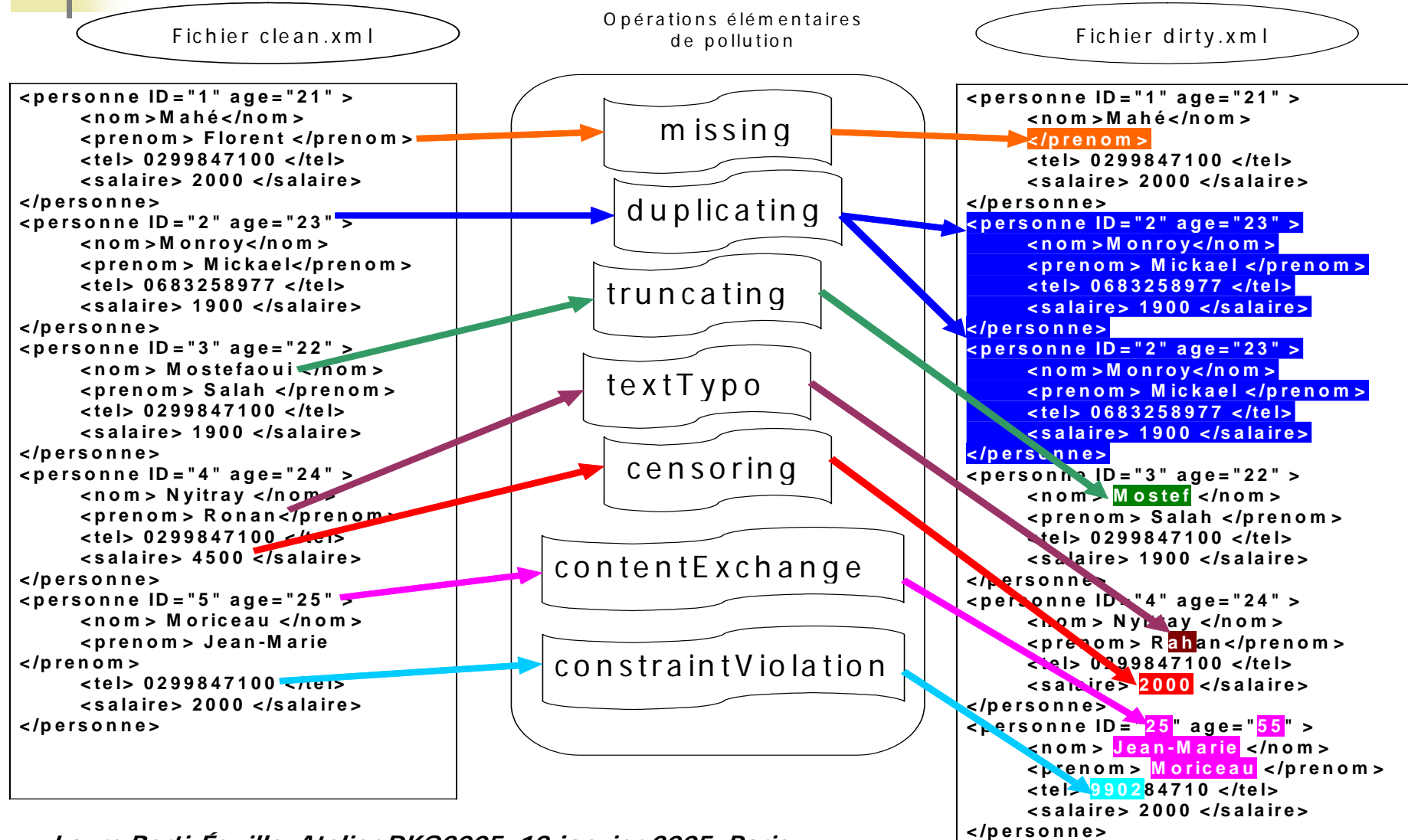
- ✓ empiriques : heuristiques, algorithmes, propriétés math. simples
- ✓ statistiques : formalismes et propriétés issus de la théorie des probabilités et des réseaux bayésiens
- ✓ basées connaissances : modèles et formalismes de représentation des connaissances

# Nettoyage des données


## Comparaison de prototypes de nettoyage des données relationnelles

<b>Potter's wheel</b> [Raman <i>et al.</i> 2001]	Détection et correction d'anomalies transformations des données ( <i>add, drop, merge, split, divide, select, fold, format</i> )	Interactivité, inférence de la structure
<b>Ajax</b> [Galhardas <i>et al.</i> 2001]	Langage déclaratif basé sur des opérateurs logiques de transformations ( <i>mapping, view, matching, clustering, merging</i> )	Séparation logique/physique, 3 algorithmes pour l'appariement
<b>Arktos</b> [Vassiliadis 2001]	Méta-modèle unique permettant de couvrir toutes les activités ETL d'un entrepôt (architecture, gestion de la qualité, etc.)	Distinction entre le méta-modèle et les niveaux conceptuel, logique et physique
<b>Intelliclean</b> [Low <i>et al.</i> 2001]	Détection et correction d'anomalies par utilisation d'une base de règles ( <i>duplicate identification, merge, purge, stemming, soundex, abbreviation</i> )	Peu scalable
<b>Telcordia's tool</b> [Caruso <i>et al.</i> 2000]	Outil paramétrable pour <i>record linkage</i> selon des fonctions de distance et d'appariement. Les règles d'appariement peuvent être générées, testées avec des techniques statistiques ou par apprentissage automatique	pré-traitement avec élimination des stop-words

# Typologie des problèmes de QDD XML



# Démarche expérimentale

- Établir un modèle de coût basé sur les probabilités de détection et de pollution en fonction des temps de mise à jour et de maintenance, notés resp.  $Cost(U)$  et  $Cost(M)$
  - Prédire, au moyen du modèle, les temps de détection et de nettoyage, notés  $Cost(D)$  et  $Cost(S)$
  - Tester le modèle sur des données dont on connaît la pollution
- 
- Produire un générateur avancé de pollutions pour des données XML
  - Étiqueter les pollutions et calculer la distance entre chaque contenu XML sain et sa version polluée

Cette distance reflète la probabilité de pollution et connaissant  $Cost(M)$  et  $Cost(U)$ , on peut estimer  $Cost(D)$  et  $Cost(S)$

# Définitions préliminaires

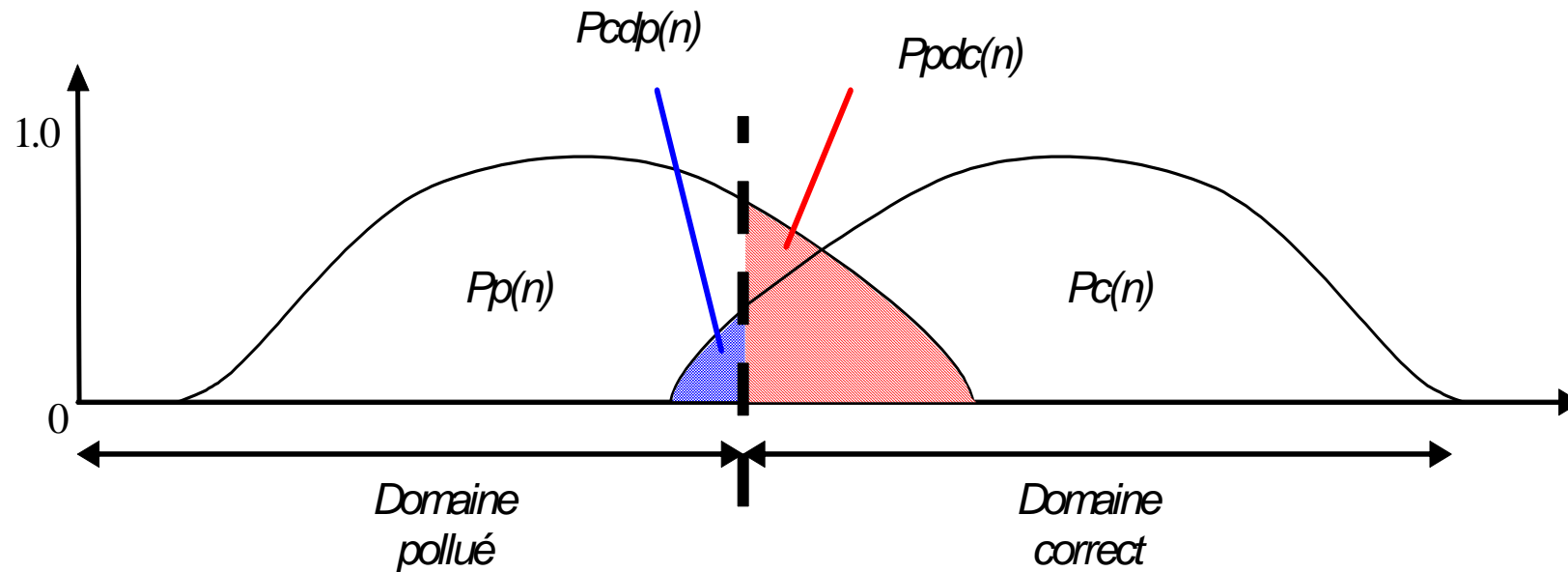
■ **Document XML pollué** : Un document XML pollué est un arbre  $T_p = (N_p, E_p)$ ,  $p \in P$ , où  
 (i) l'ensemble des noeuds  $N_p \subseteq N$  ; (ii) l'ensemble des arcs  $E_p \subseteq N_p \times N_p \times N_p$  définit un arbre racine valide par rapport au modèle de données. Au moins un noeud possède le **prédicat**  $\text{flag}^*(n, p)$  vrai indiquant qu'il contient directement ou indirectement une (ou plusieurs) fonction(s) de pollution notée  $p$ .

■ **Fonction de pollution**  $p(T, \text{nodeNumber}, [\text{minHeight}, \text{maxHeight}, N, \text{parameters}])$  a pour paramètres l'arbre  $T$ , un nombre de noeuds à polluer (qui peut être un entier ou exprimé en pourcentage), et de façon optionnelle les profondeurs minimale ou maximale localisant la région où est appliquée la pollution dans l'arbre  $T$  ou encore l'ensemble des noms des noeuds ciblés (étiquettes) et un ensemble de paramètres plus spécifiques selon le type de pollution

■ **Distance entre noeuds XML** :

$$\text{contentDistance}(n_1, n_2) = \begin{cases} \min\left(\frac{\text{infoSize}(n_1) + \text{infoSize}(n_2)}{2}, \text{qdist}(\text{text}(n_1), \text{text}(n_2))\right) \\ \text{sametag}(n_1, n_2) + \sum_c \min(\text{qdist}(\text{val}(n_1, a), \text{val}(n_2, a)), \text{attrInfo}(a)) + c_a D \\ 1 \quad \text{pour les autres cas} \end{cases}$$

# Modèle de coût proposé



$P_p(n)$  : probabilité que le nœud  $n$  soit détecté pollué,

$P_c(n)$  : probabilité que le nœud  $n$  soit détecté correct

$P_{cdp}(n)$  : probabilité que le nœud  $n$  soit détecté pollué alors qu'il est correct

$P_{pdc}(n)$  : probabilité que le nœud  $n$  soit détecté correct alors qu'il est pollué

# Coût d'une pollution

$$\begin{aligned}
 \text{Pollution } (p, n) = & A.P_{pdc} \int_{N_{clean}} P_p(n)dn + B.P_{cdp} \int_{N_p} P_c(n)dn \\
 & + E \left[ \frac{\min(\text{Cost}(U) + \text{Cost}(S_p), \text{Cost}(M))}{\text{Cost}(M)} \right] P_{pdc} \\
 & + E \left[ \frac{\min(\text{Cost}(U), \text{Cost}(M))}{\text{Cost}(M)} \right] P_{cdp}
 \end{aligned}$$

avec

$$\begin{cases}
 A = E \left[ \frac{\min(\text{Cost}(U), \text{Cost}(M))}{\text{Cost}(M)} \right] - E \left[ \frac{\min(\text{Cost}(U) + \text{Cost}(S_p), \text{Cost}(M))}{\text{Cost}(M)} \right] \\
 B = E \left[ \frac{\min(\text{Cost}(U) + \text{Cost}(D), \text{Cost}(M))}{\text{Cost}(M)} \right] - E \left[ \frac{\min(\text{Cost}(U), \text{Cost}(M))}{\text{Cost}(M)} \right]
 \end{cases}$$

$$n \text{ est pollué par une pollution de type } p \text{ si } \frac{P_p(n).P_{pdc}(n)}{P_c(n).P_{cdp}(n)} \geq \frac{B}{A}$$



# Coût d'un nettoyage

$$Detect + D(n, np).Cleaning_p \geq Update(D(n, np) + 1)$$

avec

$$\left\{ \begin{array}{l} Update = E \left[ \frac{\min(Cost(U), Cost(M))}{Cost(M)} \right] \\ Cleaning_p = E \left[ \frac{\min(Cost(U) + Cost(S_p), Cost(M))}{Cost(M)} \right] \\ Detect = E \left[ \frac{\min(Cost(U) + Cost(D), Cost(M))}{Cost(M)} \right] \\ D(n, np) = contentDistance(n, np) \end{array} \right.$$



# Conclusion et perspectives

---

- ☑ Proposition d'un modèle de coût d'un scénario de nettoyage de données XML
- ☑ Développement d'un générateur de pollutions
  - Tests en cours, retours sur le modèle et validation
  - Expérimentations à plus grande échelle (XML Benchmark)