



# Vers une plateforme de gestion des schémas XML

**Sana Sellami**  
**Aicha-Nabila Benharkat**  
**Youssef Amghar**



*Institut National des Sciences Appliquées de Lyon,  
France*

**Rami Rifaieh**



*San Diego Supercomputer Center  
San Diego, California*

# Contexte général

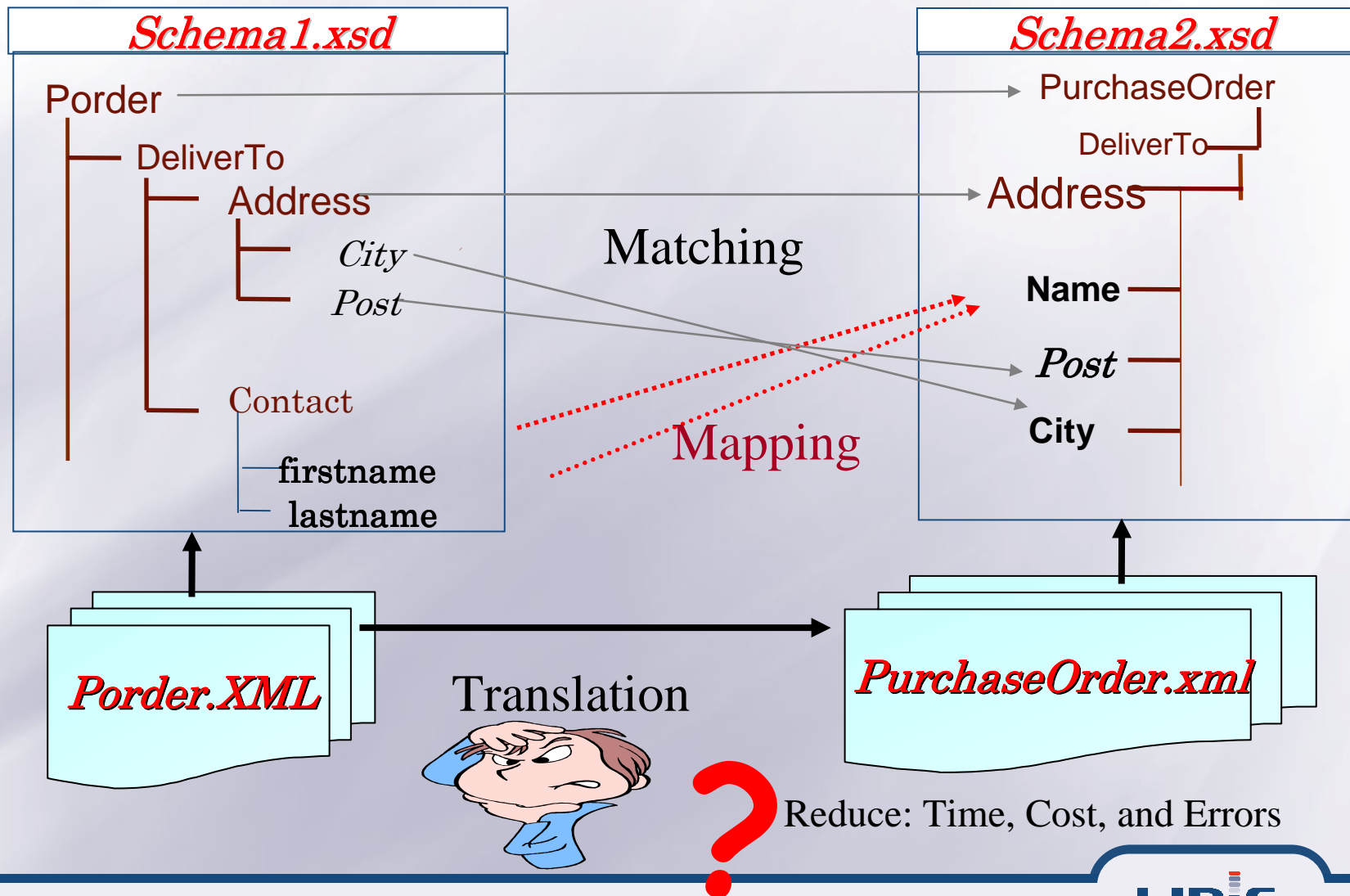
- ☰ **Caractérisation de l'environnement informationnel actuel par des données fortement distribuées.**
- ☰ **Données hétérogènes → Modèles différents pour la représentation de l'information.**
- ☰ **Intégration des données et amélioration de la transformation et de la gestion des schémas XML**



**Domaines de matching et de mapping des schémas XML**



# Contexte

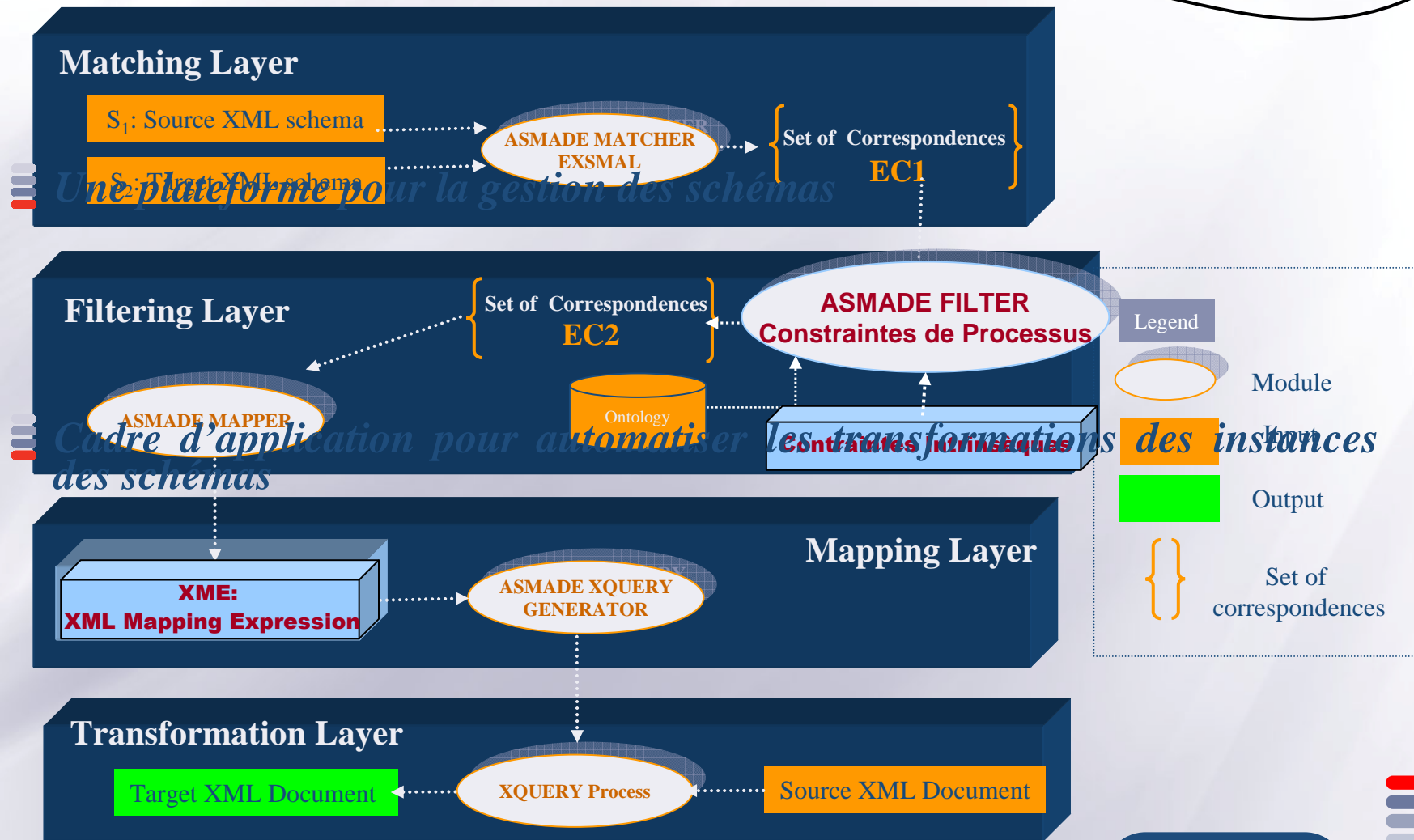


# Motivations

- *Concevoir une plateforme de gestion des schémas XML qui résiste au passage à l'échelle*
- *Intégrer les processus de matching et de mapping*
- *Etendre le processus de matching à la gestion des contraintes*
- *Proposition d'un modèle de mapping et un ensemble d'opérateurs de transformation*

# ASMADE (Automated Schema Mapping for Document Exchange)

## Contributions



# Plan

1

## *Contributions*

*Extension du processus de matching à la gestion des contraintes*

*XME: XML Mapping Expression*

2

## *Implémentation*

3

## *Conclusions et perspectives*



# Matching des schémas

- Découverte de correspondances sémantiques entre les éléments des schémas.
- Prend deux schémas de données en entrée et retourne à la fin les valeurs de similarités sémantiques entre les éléments des schémas.

# Gestion des contraintes dans le Matching des schémas

<i>Algorithmes</i>	<i>Contraintes</i>
<b>Cupid</b> [Madhavan& all, 01]	Types de données, rangs de valeurs, cardinalité, etc.
<b>Similarity Flooding</b> [Rahm & all,02]	Clés primaires, clés uniques , contraintes référentielles, cardinalité, etc.
<b>XClust</b> [Lee& all,02]	Cardinalité, compatibilité, etc.
<b>EXSMAL</b> (EDI/XML semi-automatic Schema Matching Algorithm) [Rifaieh & all, 05]	Contraintes de <i>Matching</i> : contraintes <i>Intrinsèques+Processus</i>



# *Classification des contraintes de Matching*

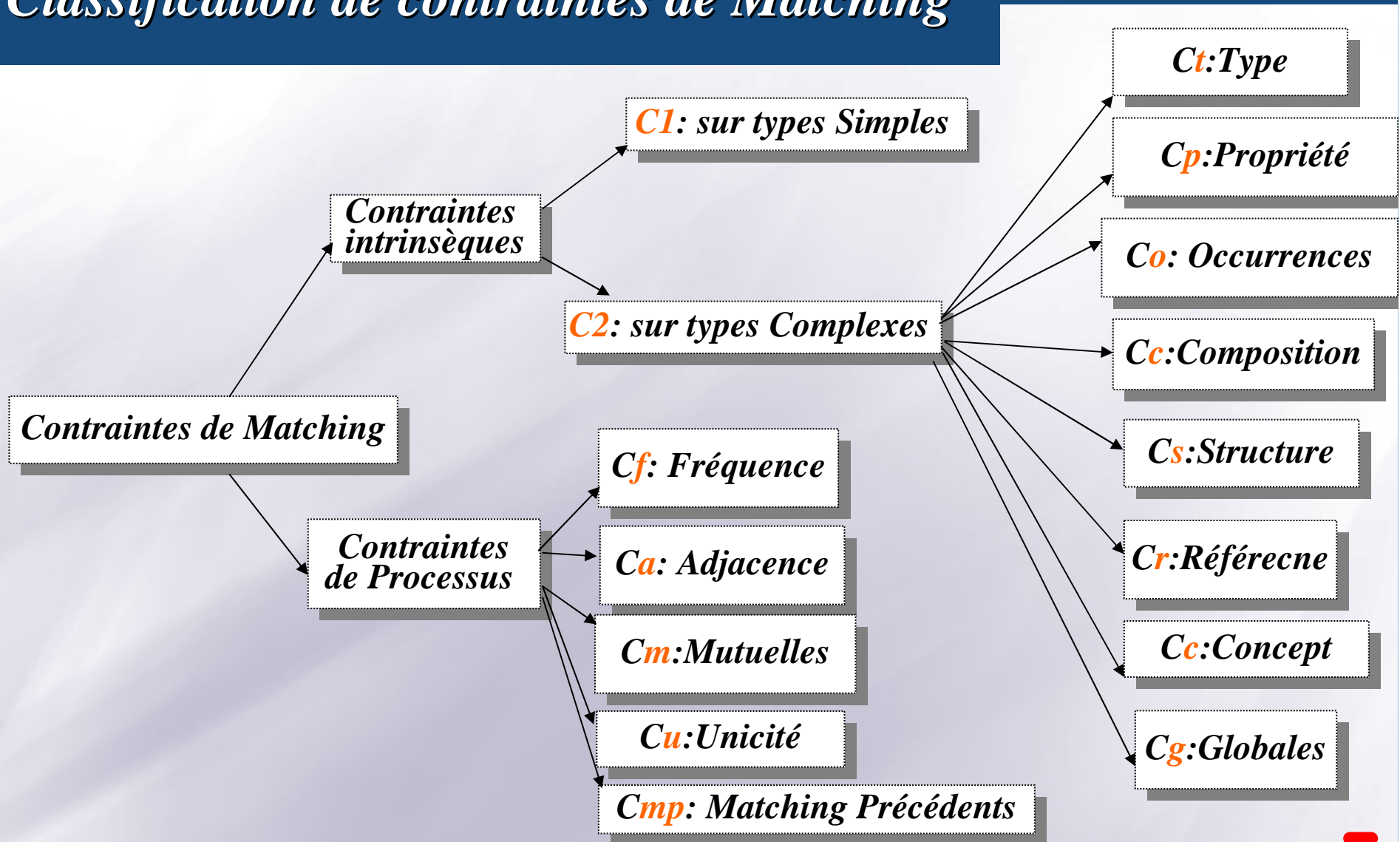
## *Contraintes intrinsèques*

- *Se rapportent aux informations liées aux éléments*
- *Classées en deux catégories: **Simple**s et **Complex**es types dans les schémas XML*

## *Contraintes de processus*

- *Spécifiées par l'utilisateur ou par les experts du domaine*
- *Détermination des correspondances sémantiques entre les schémas dans le domaine*

# Classification de contraintes de Matching



# Description de l'algorithme

## Quatre Etapes

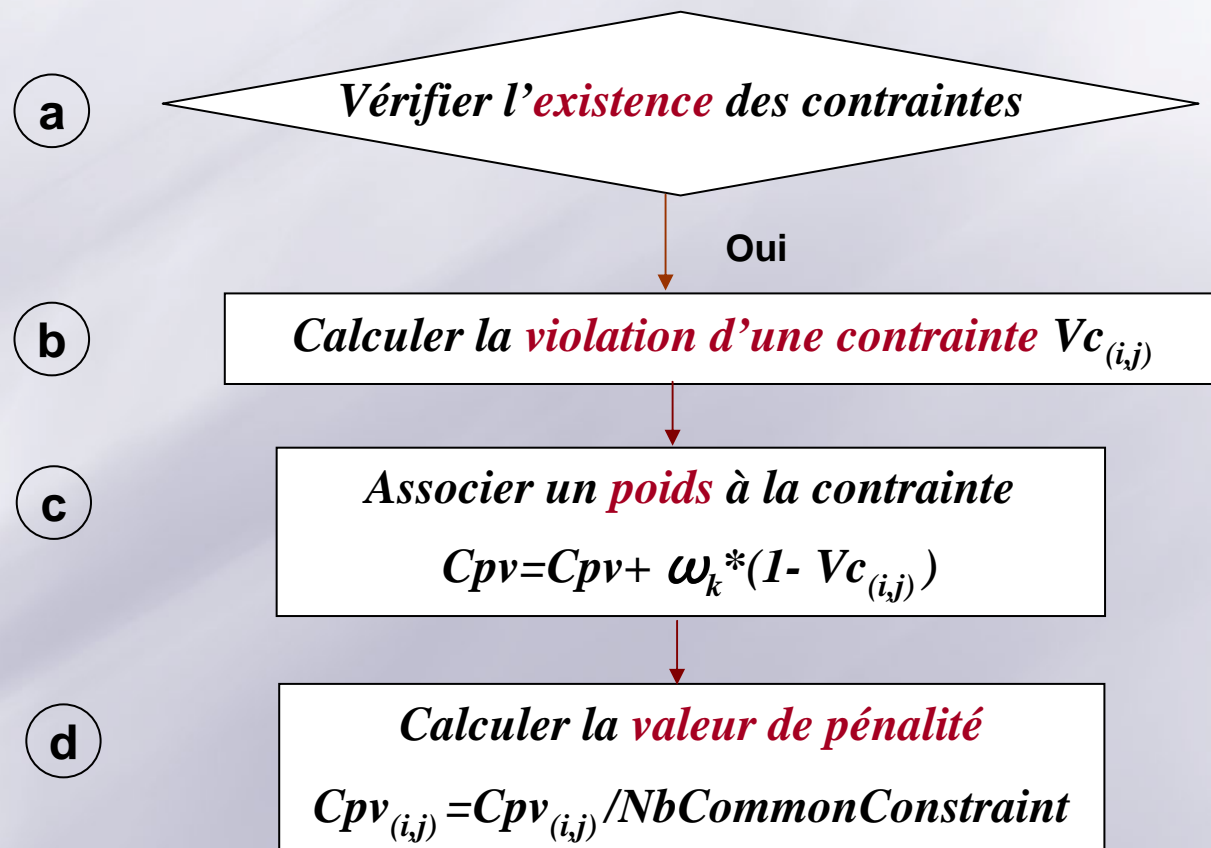
Etant donné un ensemble de similarités entre les noeuds  $e_i$  et  $f_j$ , faire:

- ① Déterminer le *type* et la *compatibilité* entre les contraintes
- ② Calculer la *pénalité de violation* des contraintes simples et/ou complexes
- ③ Calculer la *valeur finale de la pénalité de violation* des contraintes de matching
- ④ Déterminer le *nouvel ensemble de correspondances* satisfaisant les contraintes



# Description de l'algorithme

## Etape 2: Calcul de la pénalité de violation des contraintes ( $C_{pv}$ )



# *Description de l'algorithme*

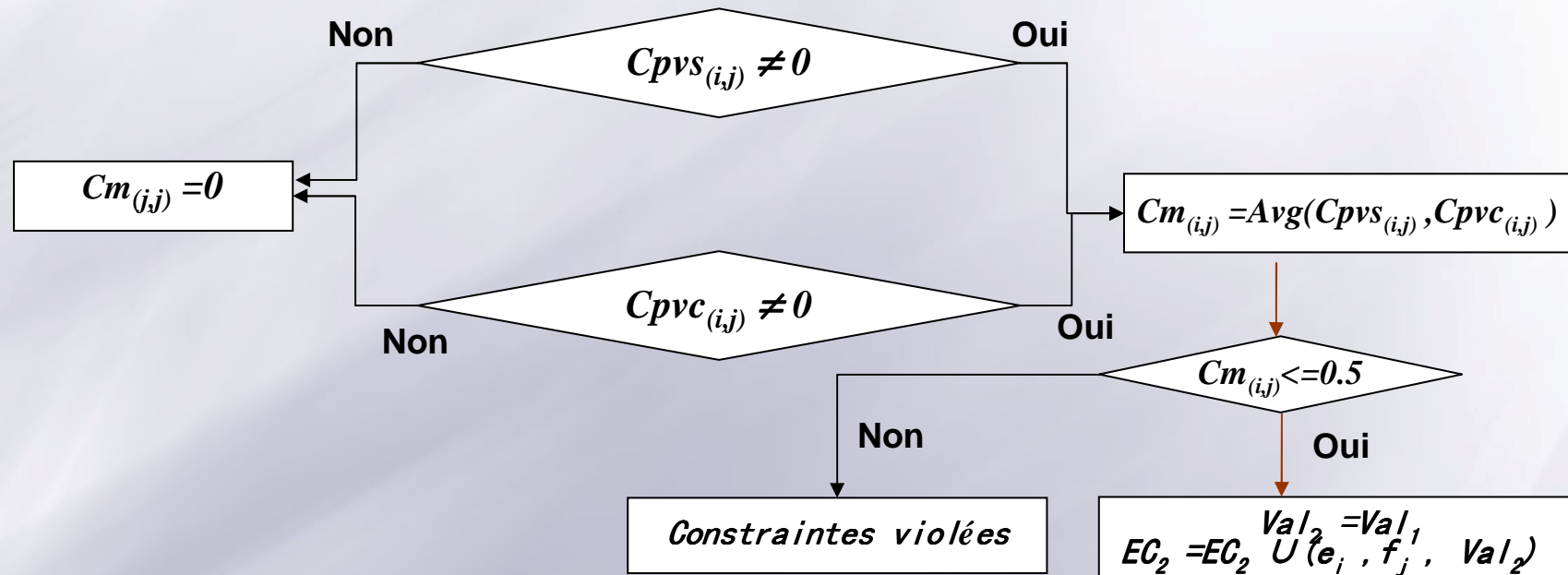
## *Etape 2<sub>b</sub> : Calcul de la violation d'une contrainte*

- 1. Déterminer les contraintes dans des **domaines numérique finis***
- 2. Déterminer la compatibilité des domaines et leur intersection*
- 3. Calculer le résultat final en fonction de la cardinalité de l'intersection par rapport à la cardinalité de chaque domaine*



# Description de l'algorithme

**Etapes 3 et 4:** Calculer la valeur finale de pénalité de violation ( $C_m$ )



$C_{pvs}$ ,  $C_{pvc}$ : Pénalité de violation des contraintes simples et complexes respectivement  
 $EC_2$ : Nouvel ensemble de correspondances qui ne violent pas les contraintes

# Plan

1

## *Contributions*

*Extension du processus de Matching à la gestion des contraintes*

*XME: XML Mapping Expression*

2

## *Implémentation*

3

## *Conclusion et perspectives*



## Définitions

- Trouver la vraie relation sémantique qui permet le passage des éléments de schémas à d'autres.
- Des expressions décrivant le moyen dont les instances du schéma cible sont dérivées à partir des instances des sources.
- Une compréhension profonde de la sémantique des schémas sources + liens sémantiques entre les schémas sources et cibles.



# Modèles de Mapping

- *Model Management System* [Bernstein, 03]
- *LIMXS* [Boukottaya, 04]
- *Value correspondences* [Miller,00]
- *Mapping Expression Model* [Rifaieh,04]

## *Limites:*

Les modèles **ne couvrent pas:**

- Mapping n:m
- Tous les cas de génération d'éléments cibles



# XME: XML Mapping Expression

## Objectif

Exploiter les correspondances pour exprimer des règles de transformation

## Expression des règles dans le modèle XME(XML Mapping Expression)

$$A_i = \sum_i f_i(a_1, a_2, \dots, a_e) L_i C_i + \tau_i = \alpha_i(A) + \tau_i$$

$\alpha_i$ : Permet de générer une partie de l'attribut/champ cible

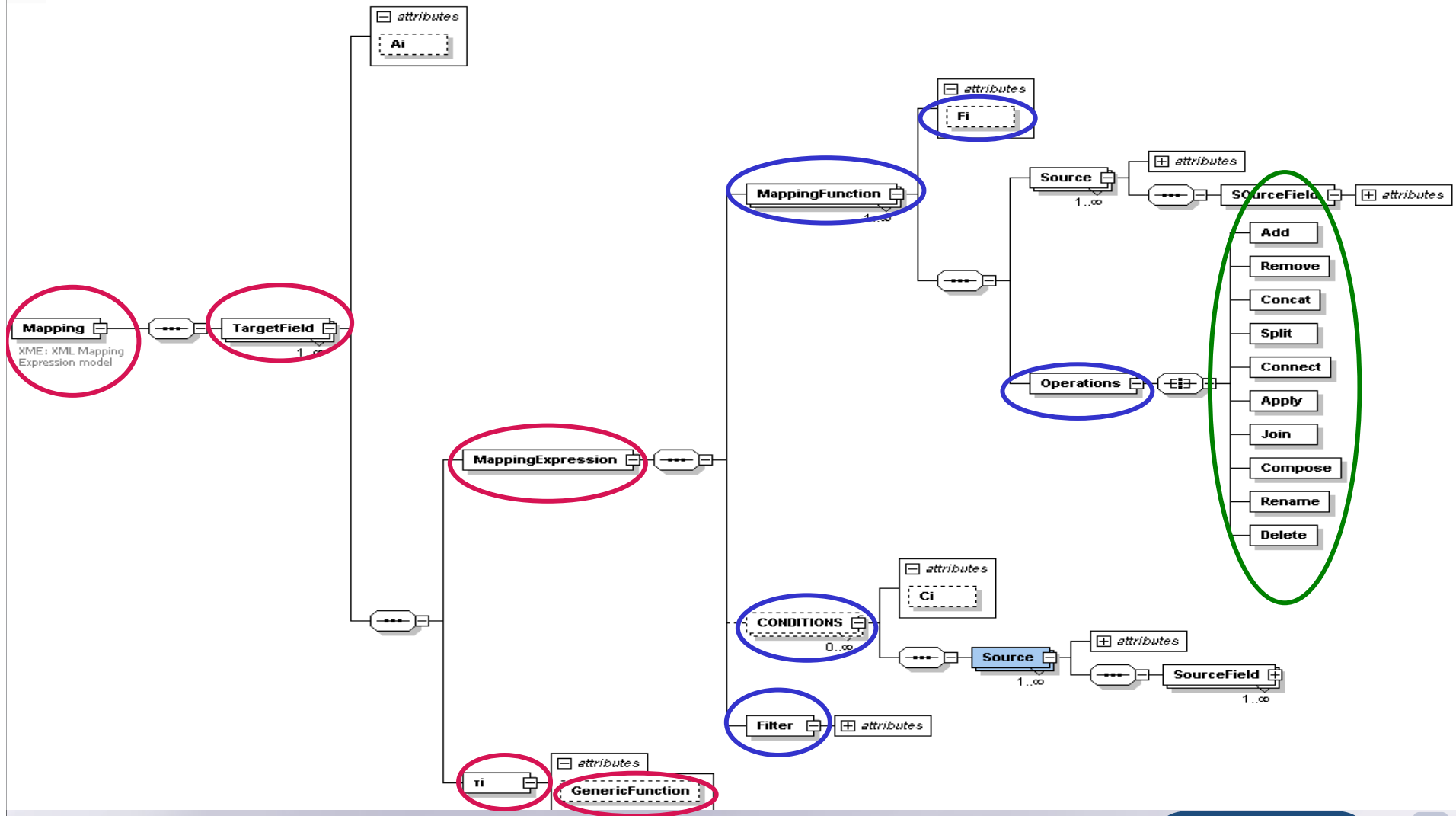
$f_i$ : Fonction de mapping

$L_i$ : Ensemble de filtres sur les données sources

$C_i$ : Condition à satisfaire avant d'effectuer le mapping

$\tau_i$ : GenericFunction: fonction générique

# Représentation de XME



# Plan

1

**Contributions**

2

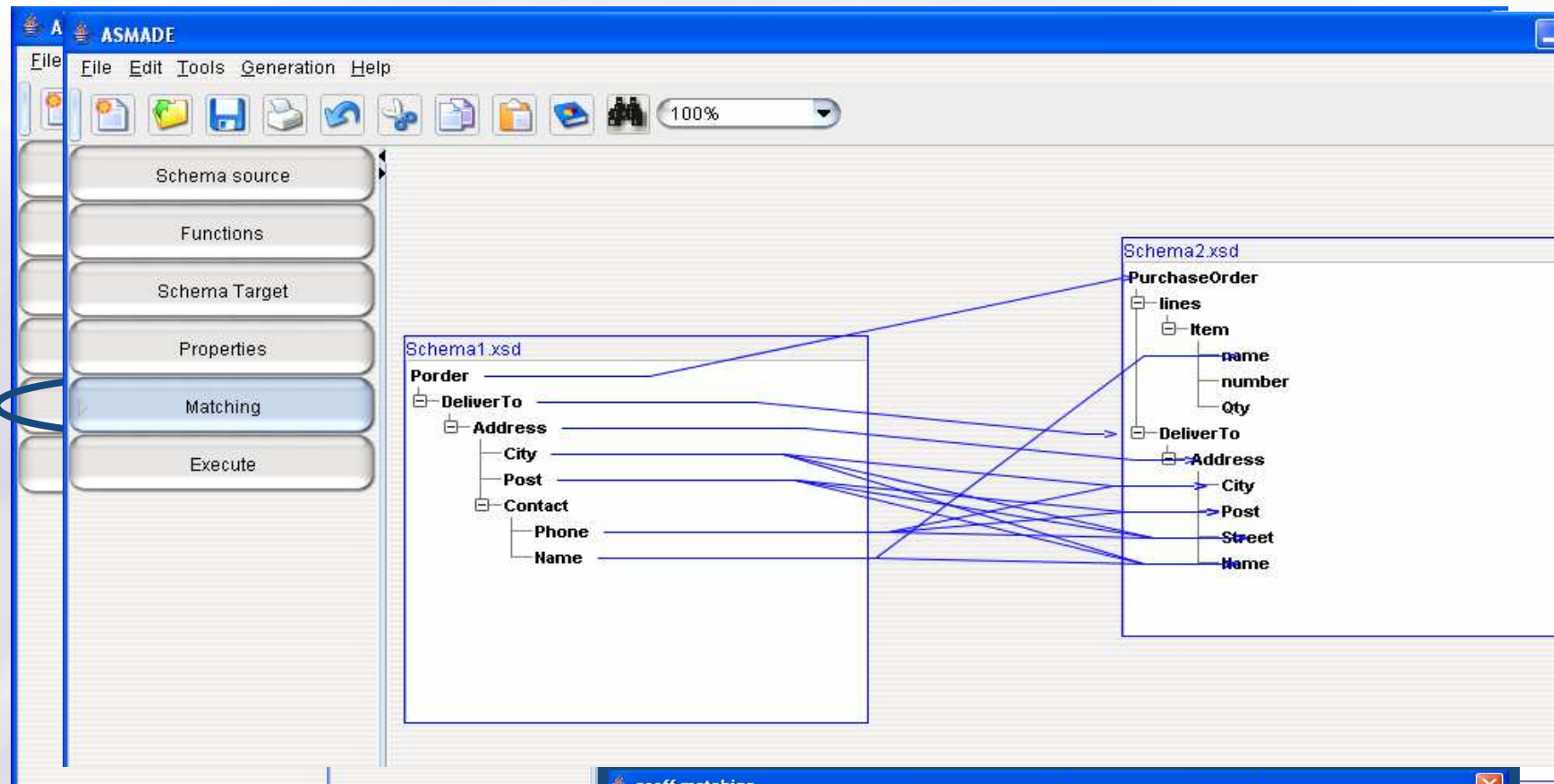
**Implémentation**

3

**Conclusion et perspectives**



# Implémentation



Description Coefficient	Ancestor Node Coefficient	Sibling Node Coefficient	Immediate Child Node Coeffi...
0.7	0.25	0.25	0.25
Standard Deviation Threshol...	Basic Similarity Coefficient	Acceptation Threshold	Match
20	0.6	0.5	

# Implémentation

The screenshot displays a software interface for schema mapping and file saving. At the top, two XML schemas are compared: **Schema1.xsd** (Perder) and **Schema2.xsd** (PurchaseOrder). The mapping shows elements like **DeliverTo**, **Address**, **City**, **Post**, **Contact**, **Phone**, and **Name** from Schema1 being mapped to corresponding elements in Schema2.

Below the mapping is a file explorer window titled **Enregistrer** (Save). The current directory is **Bureau**. A list of files is shown, including **resultat.xml**, which is circled in blue. The file type is set to **Tous les fichiers**. Buttons for **Enregistrer** and **Annuler** are visible.

At the bottom, a **coeff matching** dialog box is open. It contains several settings:

- Description Coefficient:** 0.7
- Ancestor Node Coefficient:** 0.25
- Standard Deviation Threshold:** 20
- Basic Similarity Coefficient:** 0.6

A vertical list of values (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85) is shown, with 0.65 selected. To the right, there is a **Immediate Child Node Coeffi...** field set to 0.25. Buttons for **Match** and **Save Result** are present, with **Save Result** circled in blue.

# Implémentation

The screenshot displays an XQuery editor window titled "Untitled1.xquery". The main text area contains the following XQuery code:

```
define function fomratter($fichier_donnees)
{
  for $donnees in $fichier_donnees
  return
  (
    <PurchaseOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="C:\Eclipse 3.0\workspace\asmade2\schema\Schema2.xsd">
      <lines>
        <Item>
          <name/>
          <number/>
          <Qty/>
        </Item>
      </lines>
      <DeliverTo>
        <Address>
          <City> $donnees/Porder/DeliverTo/Address/City <City/>
          <Post> $donnees/Porder/DeliverTo/Address/Post <Post/>
          <Street/>
          <Name>
            concat($donnees/Porder/DeliverTo/Address/Contact/FirstName,
              $donnees/Porder/DeliverTo/Address/Contact/LastName) <Name/>
          </Address>
        </DeliverTo>
      </PurchaseOrder>
    )
}

let $fichier := document (schema1.xml)
return
(
  writeTo(formatter($fichier), schema2.xml)
)
```

On the left, a tree view for "Schema1.xsd" shows the structure: Porder -> DeliverTo -> Address -> City, Post, Contact. On the right, a tree view for "Schema2.xsd" shows: PurchaseOrder -> lines -> Item (name, number, Qty) and DeliverTo -> Address (City, Post, Street, Name). A callout box labeled "function" points to the `concat` function in the XQuery code, with the text "Concat (LastName,FirstName)" below it.

# Plan

1

**Contributions**

2

**Implémentation**

3

**Conclusions et perspectives**

28



# Conclusion

## *Processus de Matching*

- Extension du processus de matching à la gestion des contraintes
- Classification des contraintes
- Algorithme traitant l'aspect de conformité aux contraintes

## *Processus de Mapping*

- Proposition de XME (XML Mapping Expression) et d'un ensemble d'opérateurs de transformation
- Réalisation d'un prototype de ASMADE



# Perspectives

- Intégrer l'algorithme de gestion de contraintes de matching (intrinsèques et de processus)
- Améliorer le prototype de ASMADE
- Traitement des schémas volumineux (par exemple: par fragmentation, clustering,...)
- Revoir les techniques de traitement linguistiques
- Enrichissement sémantique se traduisant par l'utilisation d'ontologies, taxonomies, thesaurus (WordNet), etc...
- Développement d'une méthodologie de partage de données résistant au passage à l'échelle



***Merci***

