Dérivation automatique d'ontologie

Université de Versailles Saint-Quentin-en-Yvelines
Orange Labs

Ivan Bedini

Pr. Georges Gardarin - Laboratoire PRiSM, Université de Versailles, Directeur de thèse

Dr. Benjamin Nguyen - Laboratoire PRiSM,, Université de Versailles, Co-directeur

Dr. Thierry Bouron - Orange Labs , Co-directeur

15 Janvier 2010, Soutenance de thèse

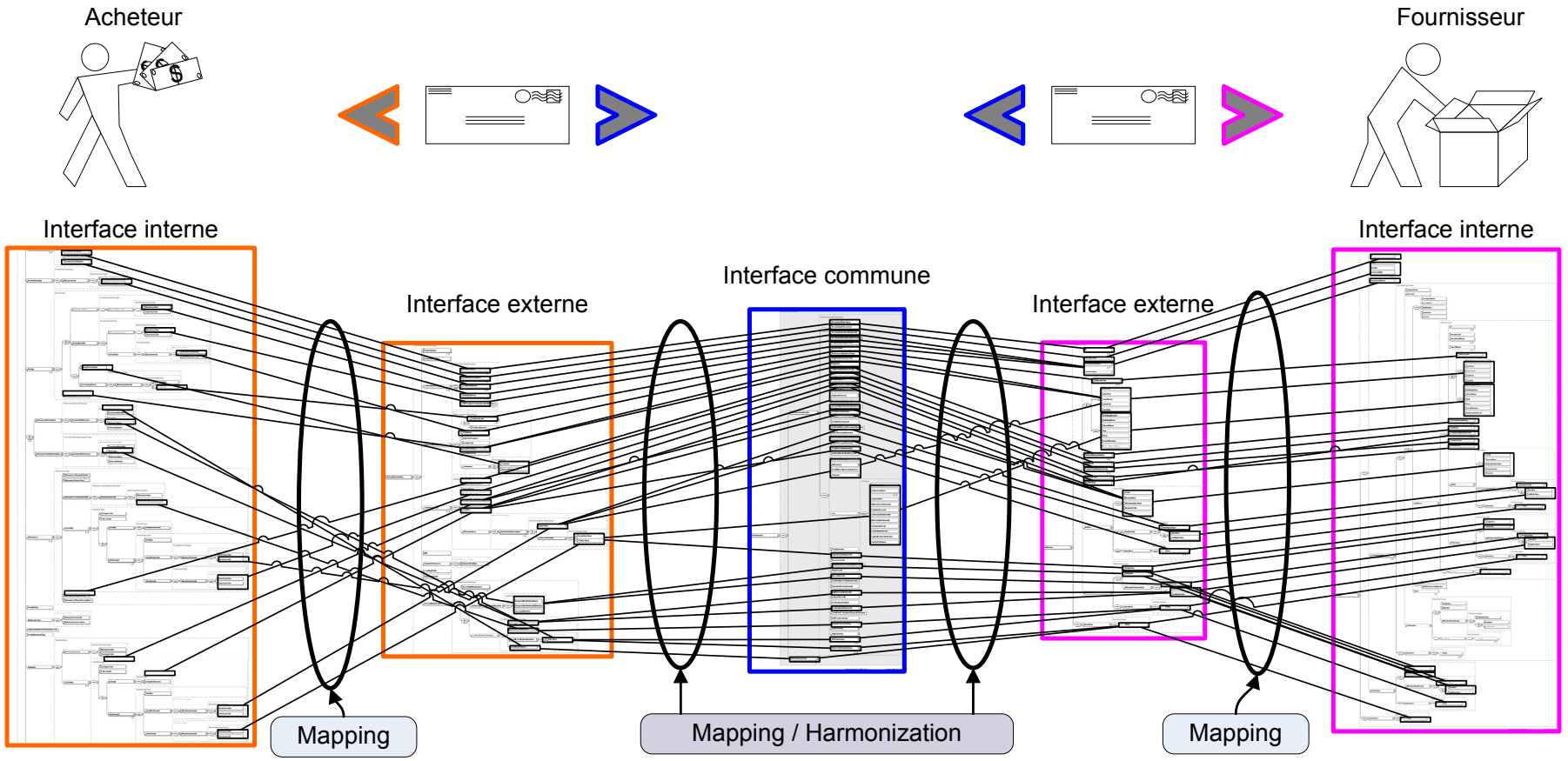
orange

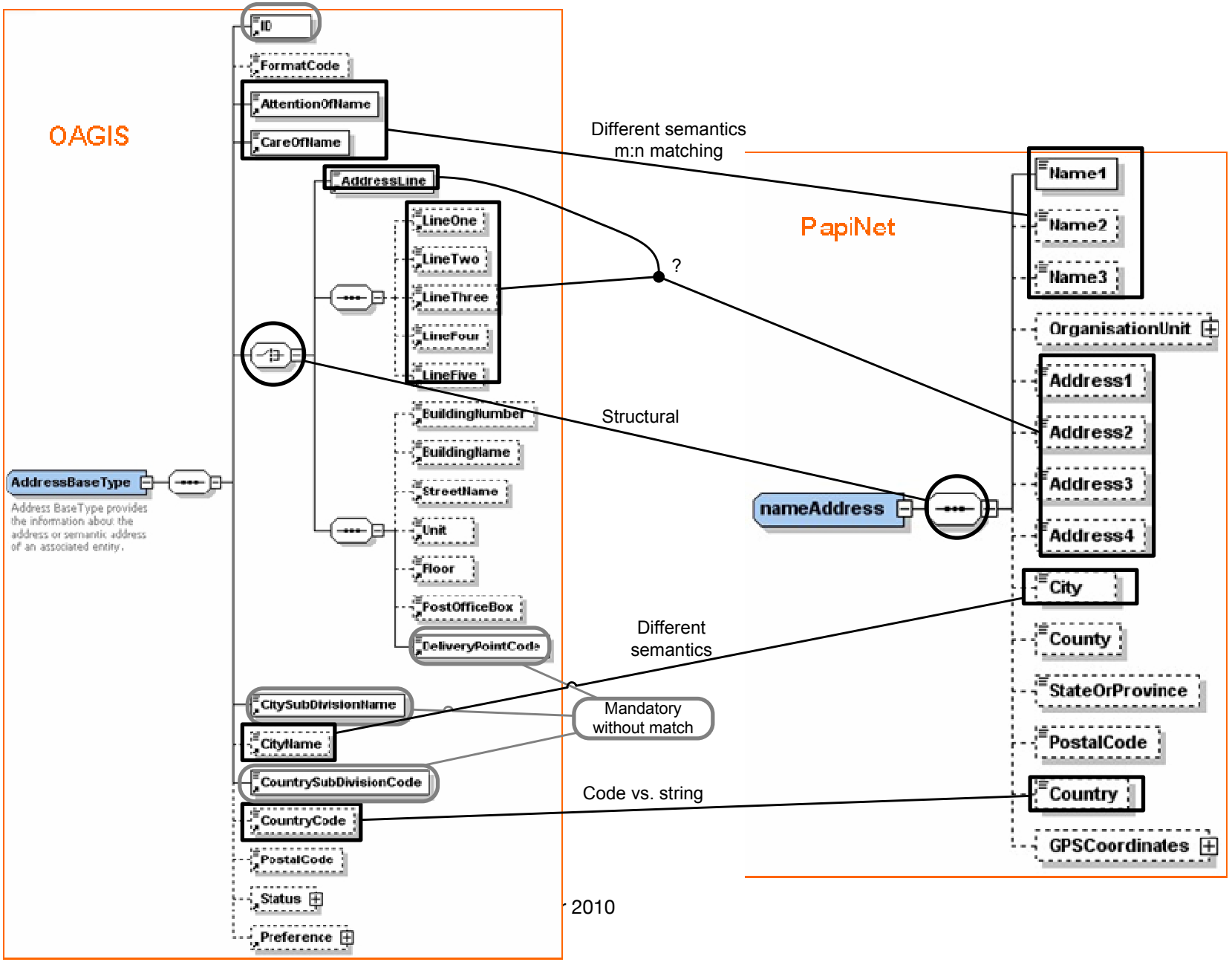
sommaire

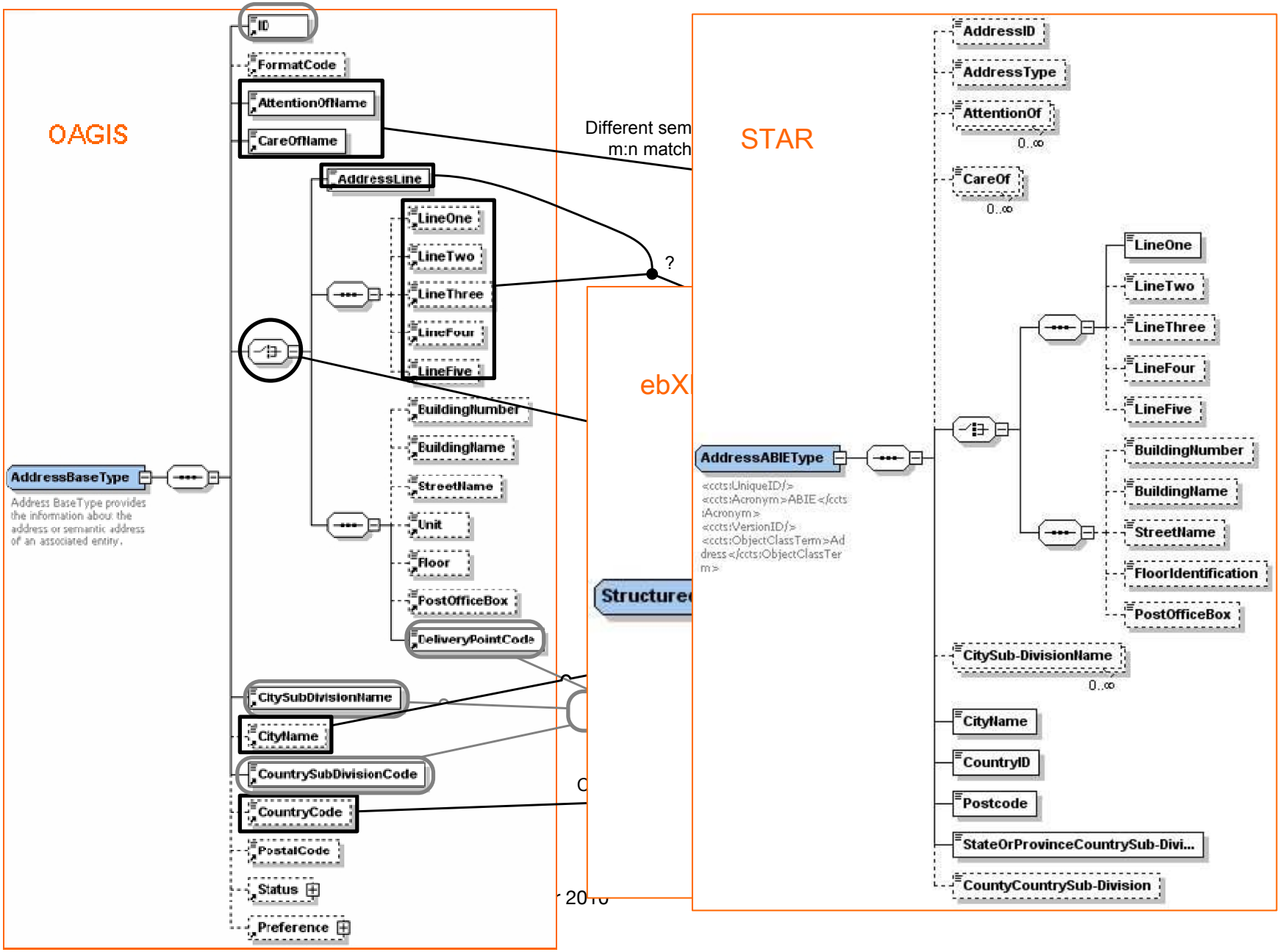
1.1

Les échanges B2B : problèmes et limitations

Scénario classique d'un échange B2B




OAGIS
ID
FormatCode
AttentionOfName
CareOfName
AddressLine
LineOne
LineTwo
LineThree
LineFour
LineFive
BuildingNumber
BuildingName
StreetName
Unit
Floor
PostOfficeBox
DeliveryPointCode
AddressBaseType
Address BaseType provides the information about the address or semantic address of an associated entity.
CitySubDivisionName
CityName
CountrySubDivisionCode
CountryCode
PostalCode
Status
Preference
Different semantics m:n matching
?
Structural
Different semantics
Mandatory without match
Code vs. string
PapiNet
Name1
Name2
Name3
OrganisationUnit
Address1
Address2
Address3
Address4
nameAddress
City
County
StateOrProvince
PostalCode
Country
GPSCoordinates


OAGIS
ID
FormatCode
AttentionOfName
CareOfName
AddressLine
LineOne
LineTwo
LineThree
LineFour
LineFive
BuildingNumber
BuildingName
StreetName
Unit
Floor
PostOfficeBox
DeliveryPointCode
AddressBaseType
Address BaseType provides the information about the address or semantic address of an associated entity.
CitySubDivisionName
CityName
CountrySubDivisionCode
CountryCode
PostalCode
Status
Preference
Different sem
m:n match
?
ebX
Structure
STAR
AddressID
AddressType
AttentionOf
0..∞
CareOf
0..∞
LineOne
LineTwo
LineThree
LineFour
LineFive
BuildingNumber
BuildingName
StreetName
FloorIdentification
PostOfficeBox
AddressABIEType
<ccts:UniqueID/>
<ccts:Acronym>ABIE</ccts:Acronym>
<ccts:VersionID/>
<ccts:ObjectClassTerm>Address</ccts:ObjectClassTerm>
CitySub-DivisionName
0..∞
CityName
CountryID
Postcode
StateOrProvinceCountrySub-Divi...
CountyCountrySub-Division

Spécification des normes B2B

- Nous avons analysé plus de **40 standards**
- Les documents définis sont pour la plupart :
  - *Dictionnaire de données, Messages, Description des interfaces de Services Web, processus d'affaire, liste de codes et messages EDIFACT*
- Formats :
  - De facto Schéma XML (XSD).
  - EDIFACT est de moins en moins ciblé dans les nouveaux projets
  - Aucun consortium ne propose encore d'ontologie

Énoncé du problème

A partir d'un grand nombre de schémas XML il est possible de **déduire automatiquement** une représentation sémantique commune qui:

i) **améliore les performances** des systèmes d'alignement et d'identification des correspondances

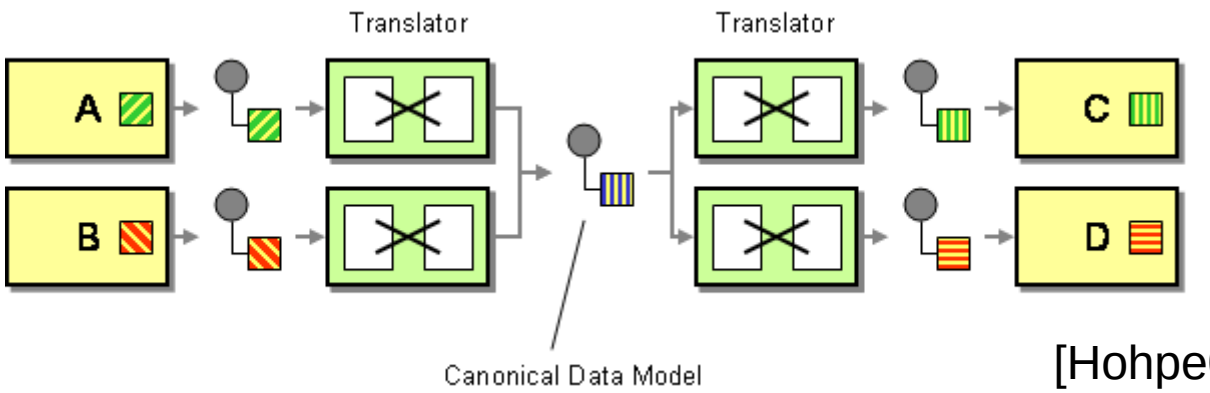
ii) peut être réutilisée pour générer **dynamiquement des ontologies**

1.2

Adoption des Ontologies : quoi et comment

Approche

- Hohpe suggère d'introduire un *Canonical Data Model* afin de minimiser les dépendances lors de l'intégration des applications, mais aucun CDM n'est encore formalisé.



[Hohpe03]

- **Nous proposons de générer des ontologies servant de modèles canoniques**

Ontologie : plus formellement

- Une ontologie *o* est au moins un *5-tuple*

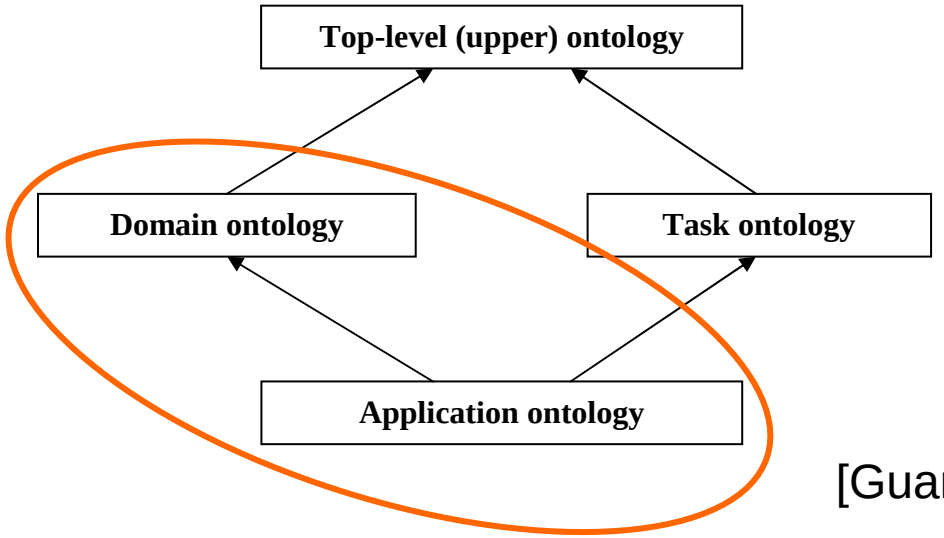
$$o = < C, R, I, D, \subseteq >$$

  - *C* est l'ensemble des **classes** ou des concepts
  - *R* est un ensemble de **relations**
  - *I* est l'ensemble d'**instances** de classes (également appelé les individus)
  - *D* est l'ensemble des **types de données**
  - $\subseteq$ est une relation binaire sur les entités appartenant à *C, R* et *D*, appelée ***spécialisation***

- **Axiomes** : des affirmations sous une forme logique qui forment ensemble la théorie générale que l'ontologie décrit dans son domaine d'application.
  - Fournisseur ≡ Entreprise ⊓ aClient.Entreprise

Ontologie : plusieurs types

- 4 niveaux d'ontologie

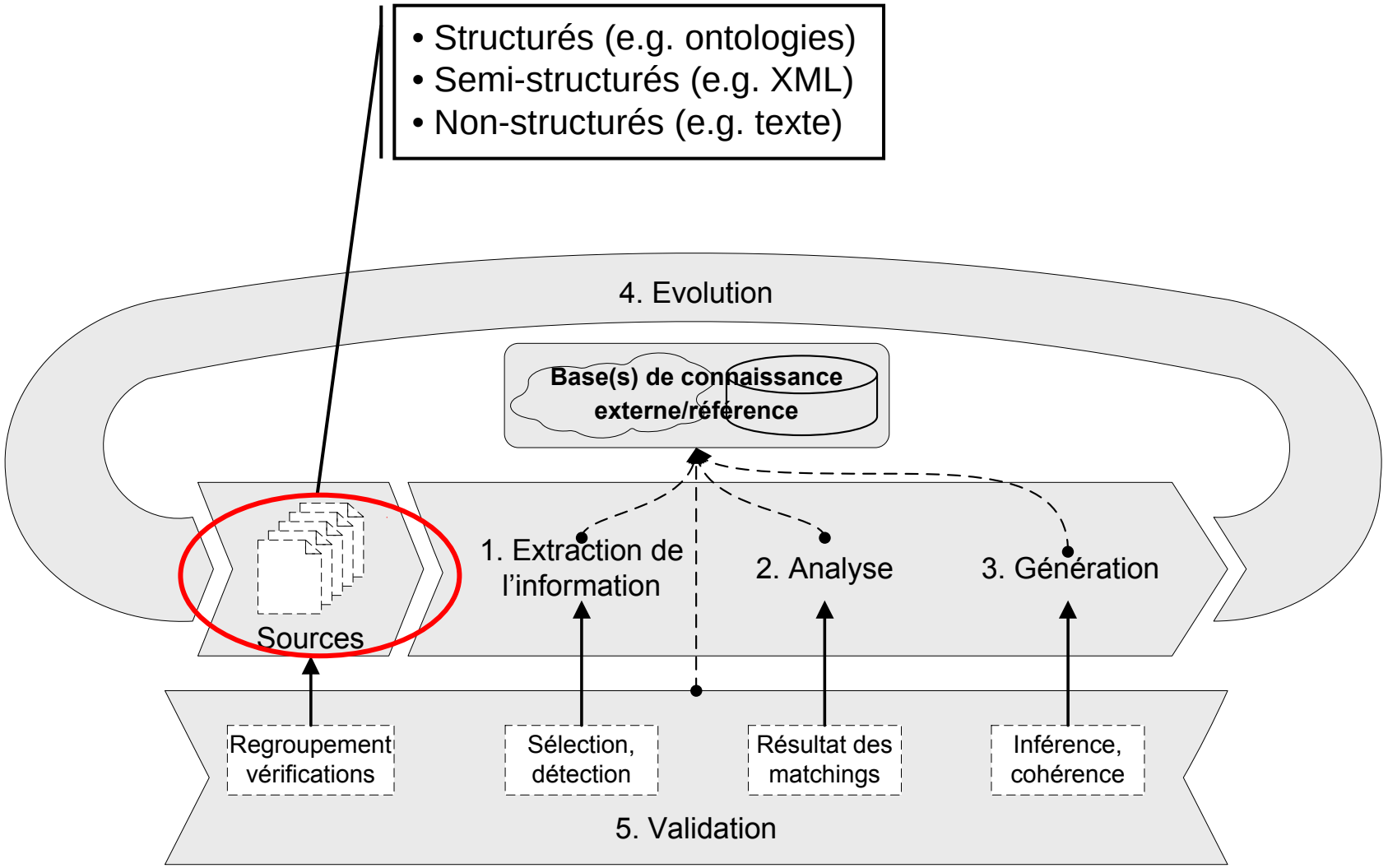


[Guarino98]

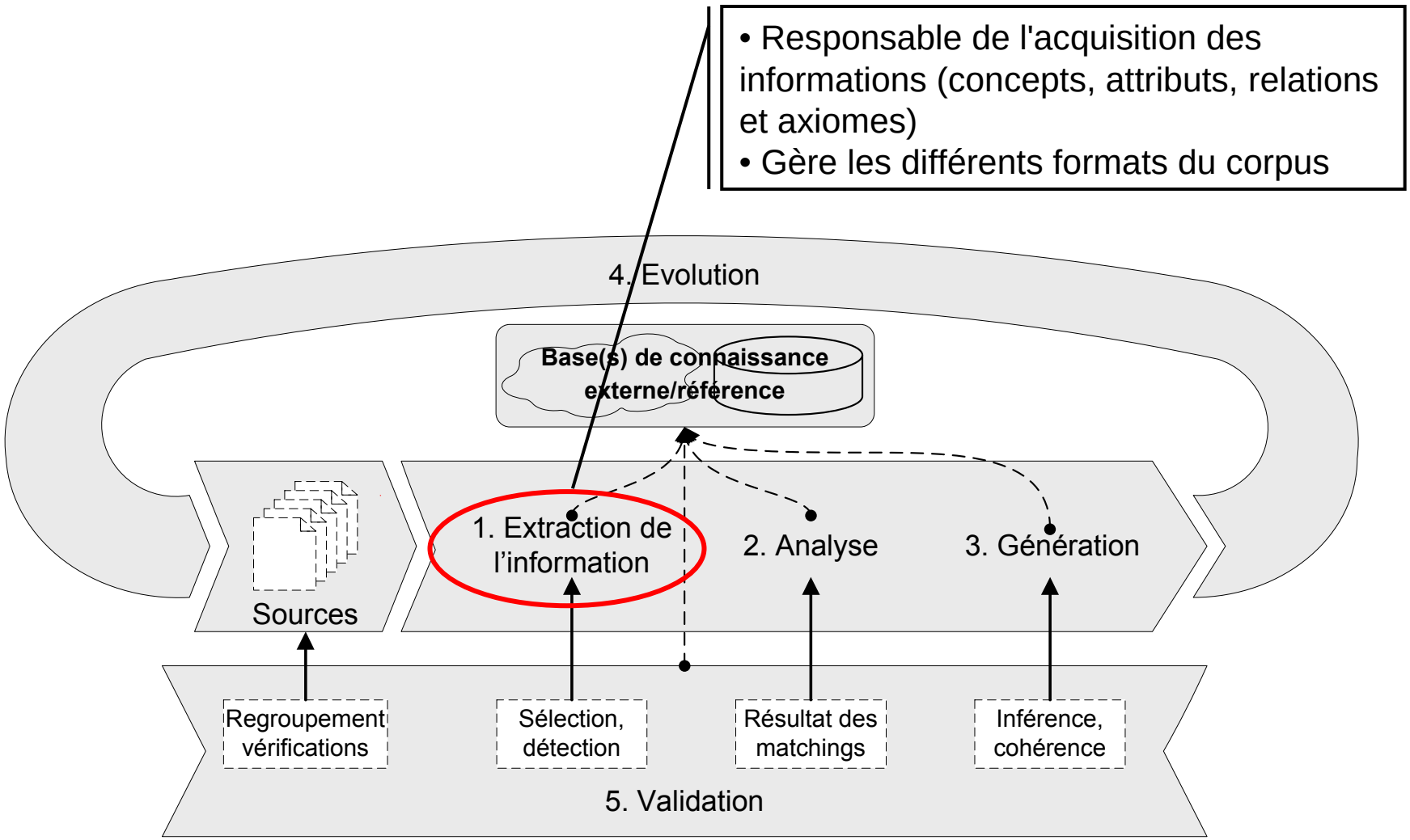
2.1

Analyse des systèmes de génération automatique d'ontologie

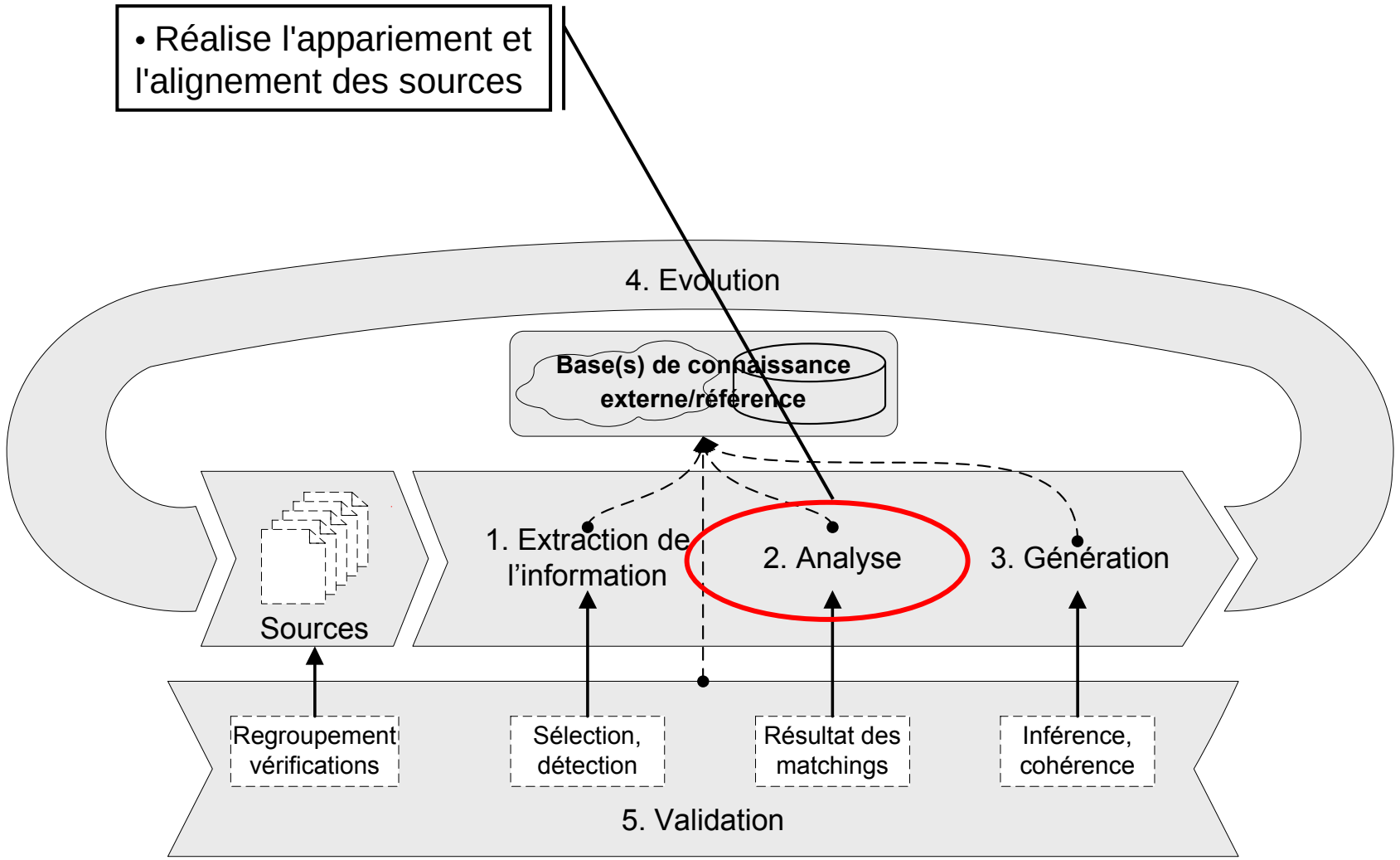
Génération automatique d'ontologie : cycle de vie



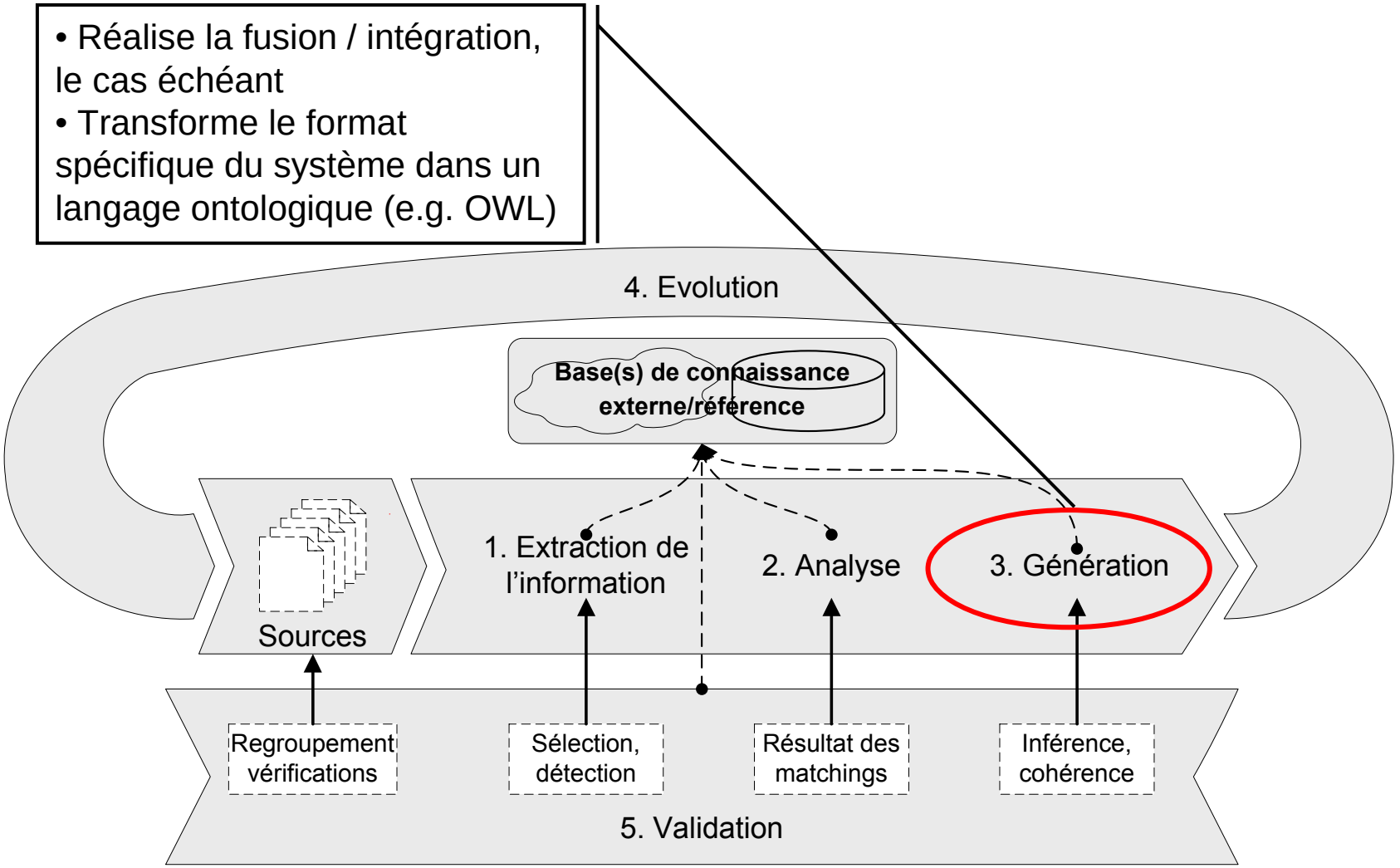
Génération automatique d'ontologie : cycle de vie



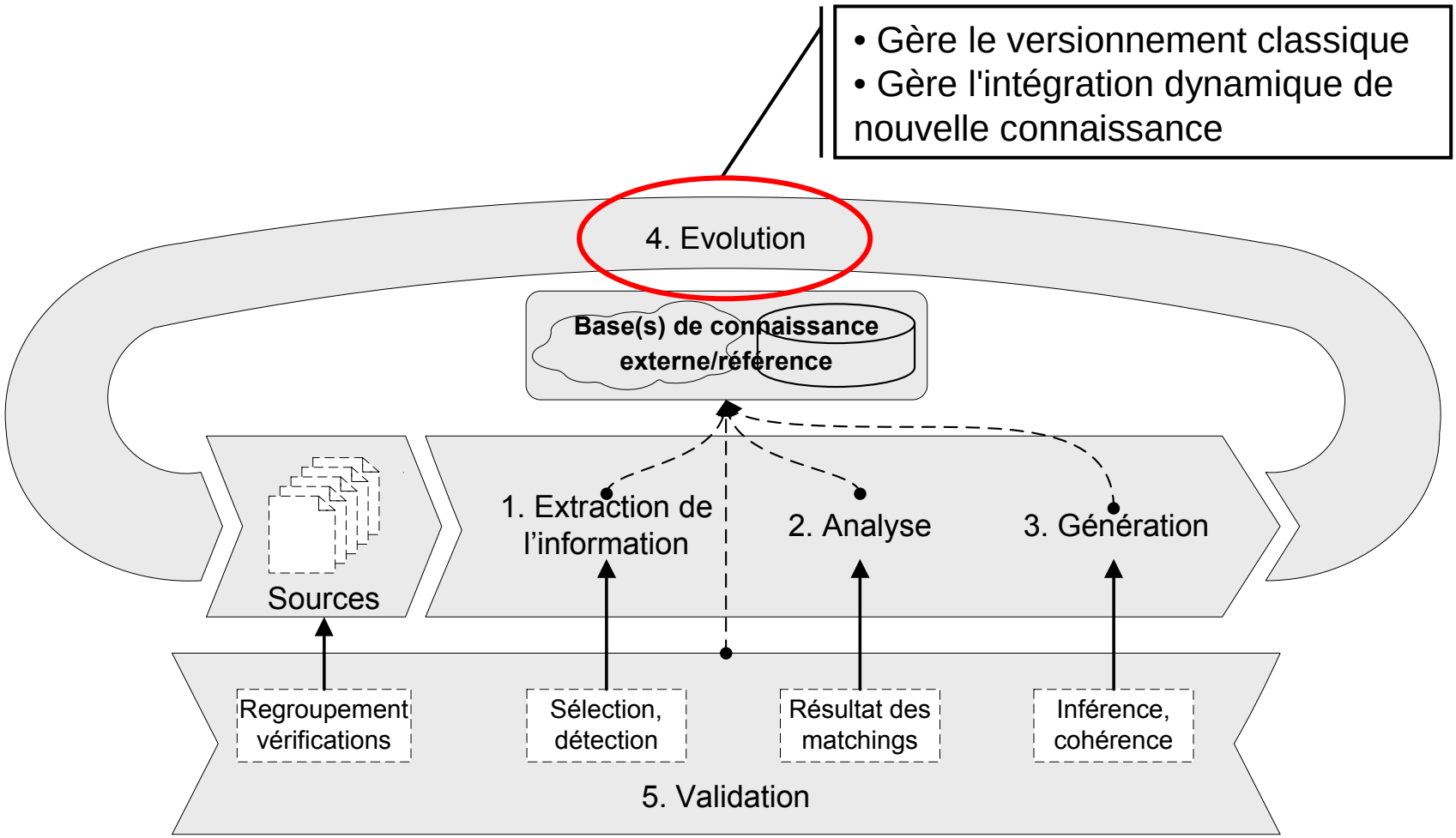
Génération automatique d'ontologie : cycle de vie



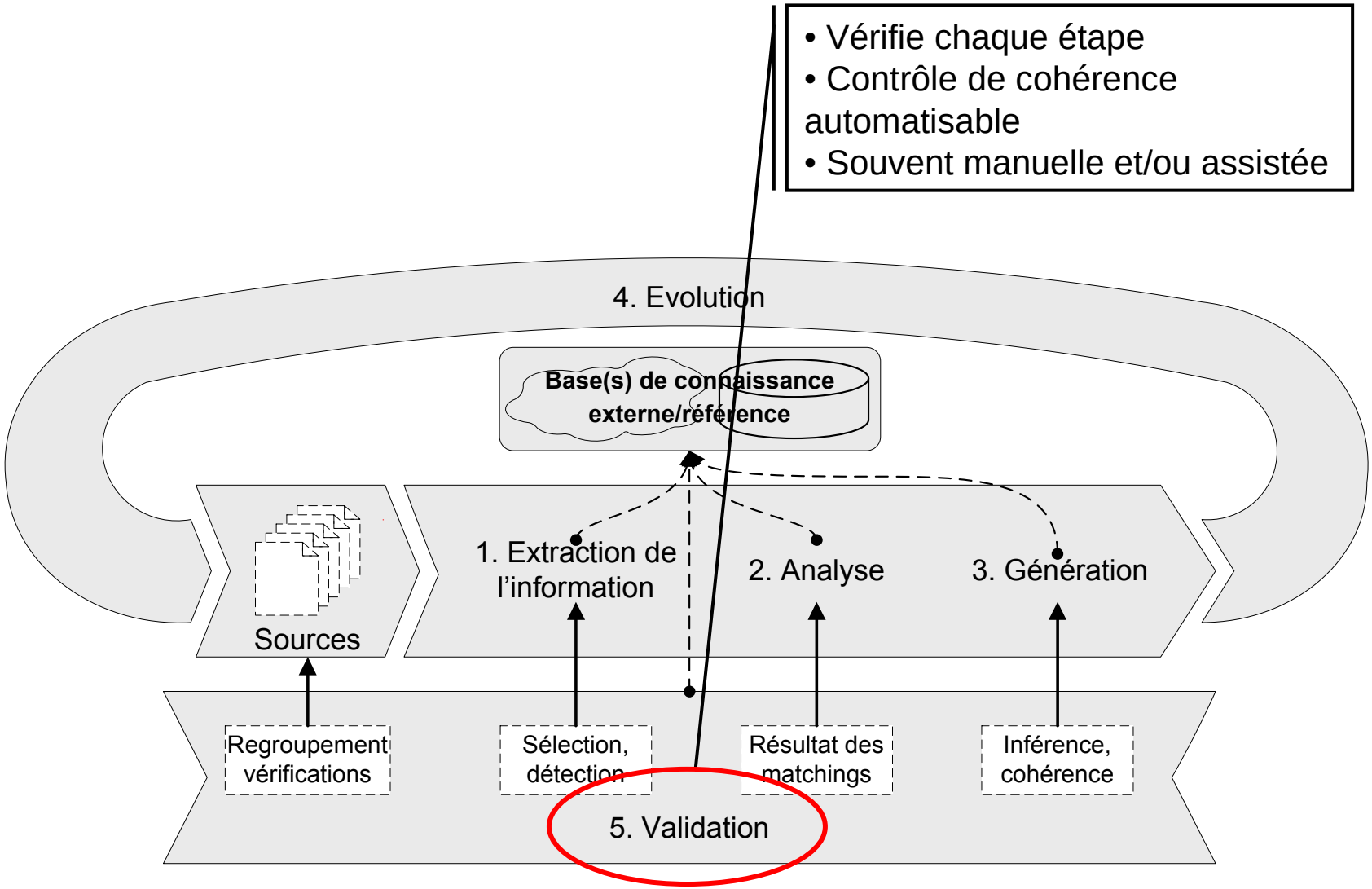
Génération automatique d'ontologie : cycle de vie



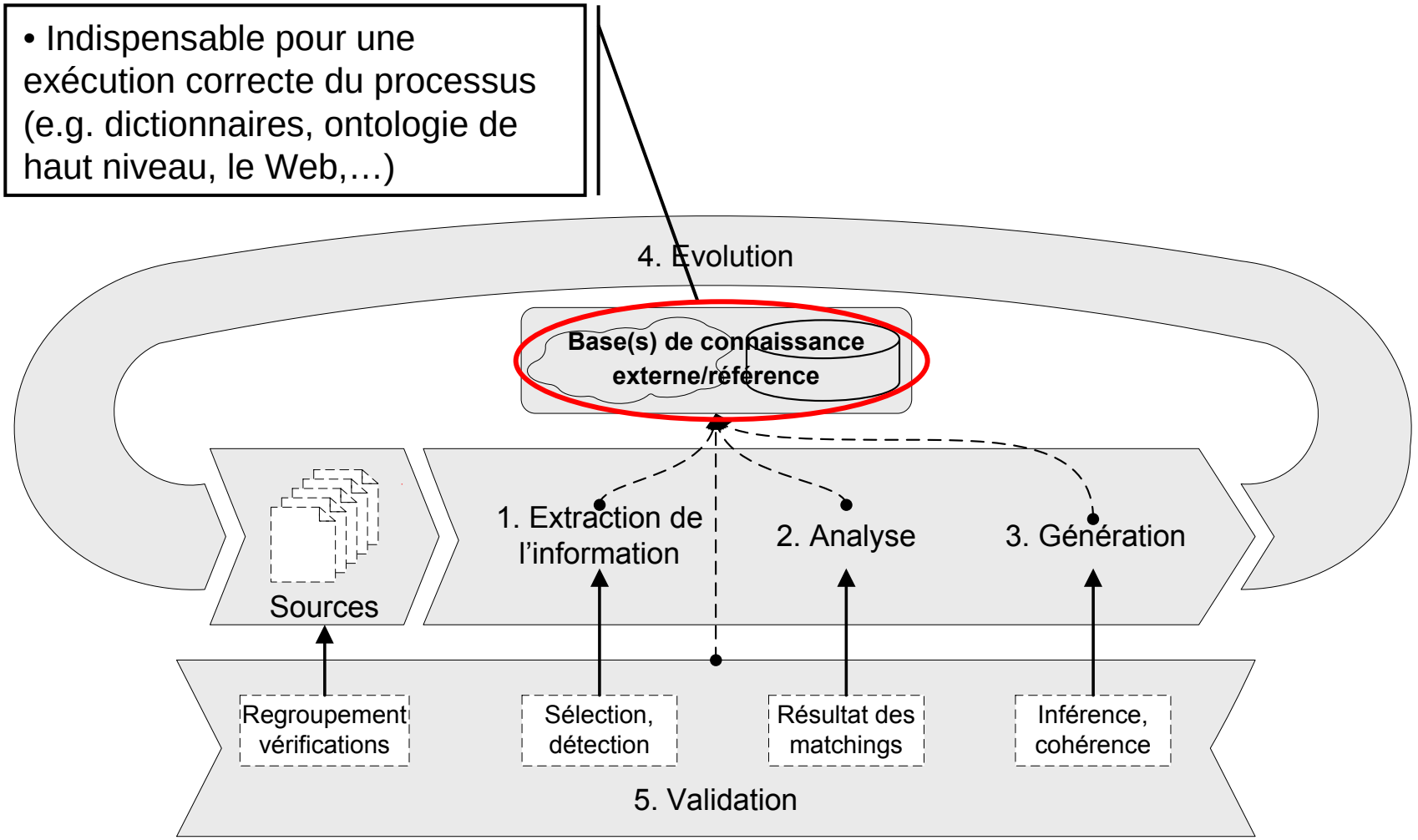
Génération automatique d'ontologie : cycle de vie



Génération automatique d'ontologie : cycle de vie



Génération automatique d'ontologie : cycle de vie

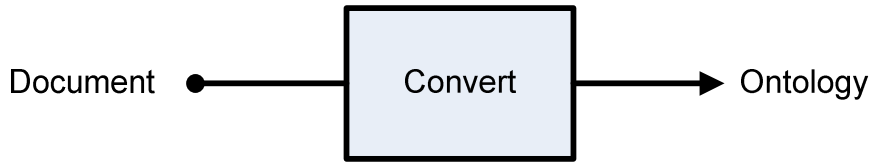


Analyse de systèmes existants

- 21 systèmes centrés sur la génération automatique d'ontologies :
  - Définir des axes de catégorisation de ces outils
  - Évaluer les systèmes selon *toutes* les étapes du cycle de vie
  - Évaluer le degré d'automatisation des outils
  - Trouver la meilleure solution pour notre cas d'utilisation

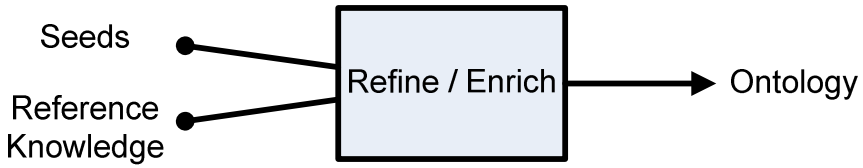
Catégorisation des systèmes

- **Conversion ou traduction directe**



  - Entrée : documents structurés (e.g. XML, XSD, UML, etc.)
  - Sortie : langage ontologique (e.g. OWL)
  - Haut niveau d'automatisation mais limitée à la tâche de génération

- **Ressource Externe**



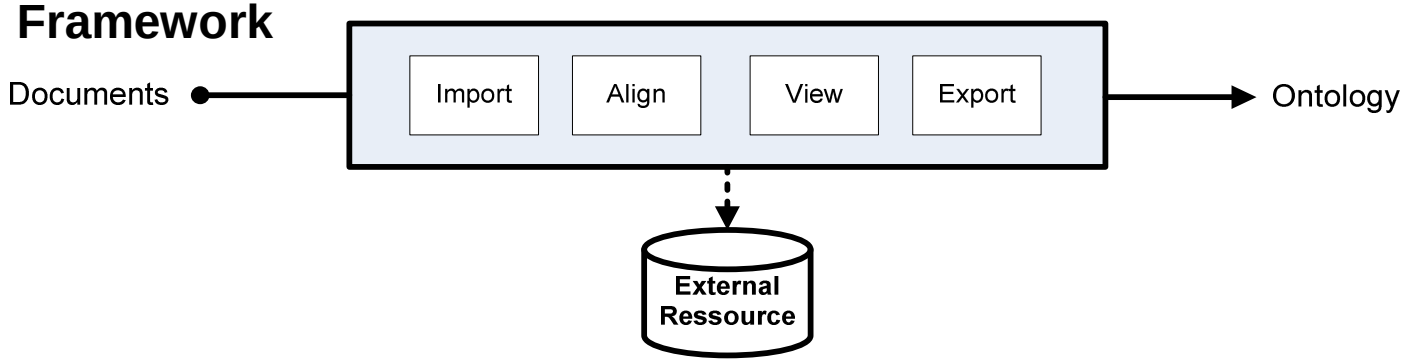
  - Problèmes :
    - Trop lié à la ressource adoptée
    - Ontologies supérieures pas suffisamment détaillées

Catégorisation des systèmes

- **Modèle intermédiaire**

  - Avantage : Approche flexible par rapport au nombre de sources
  - Inconvénient : Double transformation et risque de perte d'informations non gérées par le modèle

- **Framework**

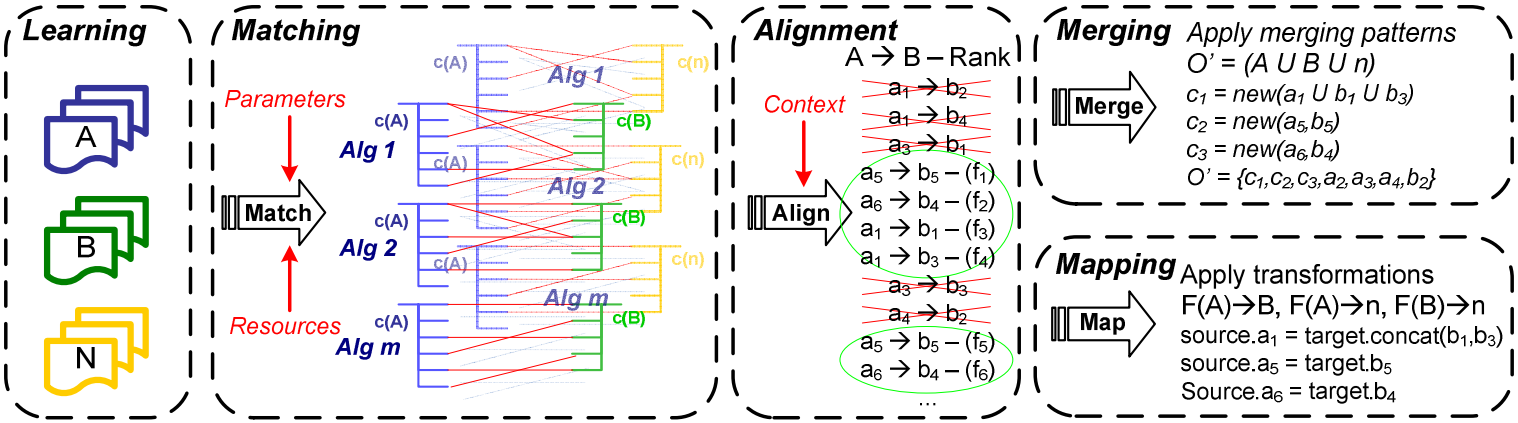


  - Avantage : Permet de gérer toutes les étapes du cycle de vie
  - Inconvénient : L'intégration des modules reste souvent humaine

Observations générales sur l'automatisation

- **Sources** multiples en entrée est une question ouverte
- **Extraction** produit des résultats exploitables, mais actuellement focalisée sur les sources textuelles
- ***Matching*** et **alignement** sont les plus difficiles à réaliser
- **Evolution** rarement implémentée et pas dynamique
- **Validation** principalement humaine
- **Ressources externes** sont rares et pas adéquates
  - **WordNet** ressource essentielle (dictionnaire, thesaurus, relations, référence, …)
  - **Web** intégration complexe et inexploitable

Le *Matching process* (opération d'appariement)



- L'opération d'appariement regroupe l'ensemble des fonctions qui recherche les correspondances entre deux ou plusieurs sources d'entrée.

  ➔ **Problème** : la fonction de recherche de correspondances agit sur un couple d'entités à la fois, indépendamment du nombre des sources à comparer

  ➔ **Conséquence** : C'est une des causes majeures du temps de calcul

  ➔ **Idée** : C'est une opération générique, donc réutilisable, dans le processus global de génération d'ontologies → optimisation possible

2.2

Définition d'un Modèle Sémantique Intermédiaire (SDMO)

Semantic Data Model for Ontologies (SDMO)

- SDMO : modèle dynamique orienté objet enrichi de méta-associations spécifiant la nature des similarités
  - Préserve les informations propres au processus de *matching*
  - Fournit une interface de détection des correspondances efficace
  - Fournit une connaissance ontologique
- Traduction complète du modèle vers OWL
- Expressivité de l'ontologie déductible : *S*HO*I*N*F*(D)

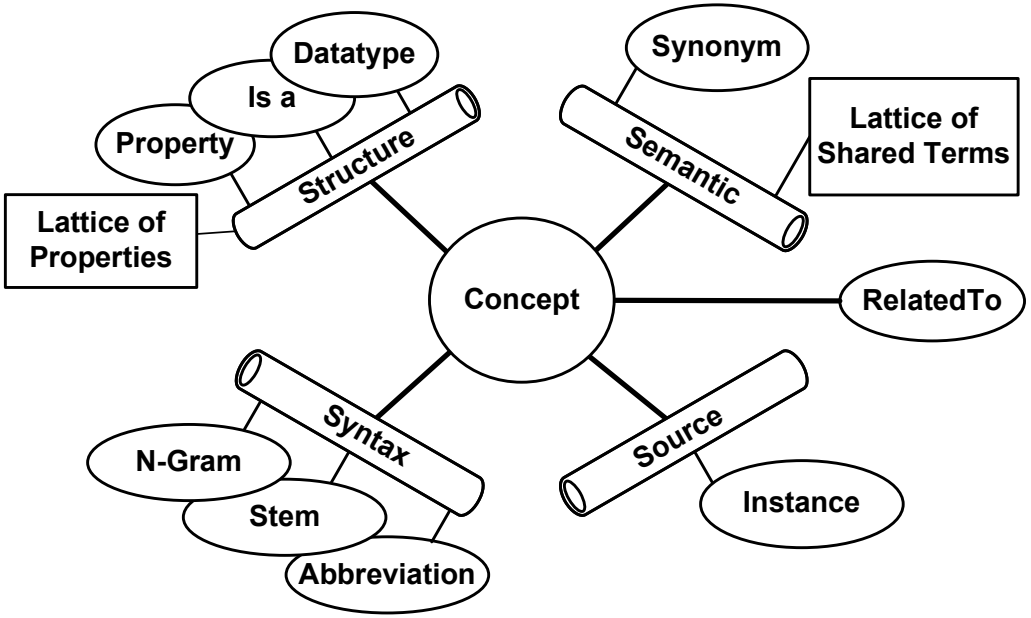
Déf. : (**Concept SDMO**)
Un concept est l'élément de base SDMO et se définit comme un 4-tuple

$$c = < I, R, S, f >$$

• ***I*** est un ensemble de mots, simples ou composés, qui représente le mieux le nom du concept.
• ***R*** est l'ensemble des relations et métarelations entre les concepts
• ***S*** est l'ensemble des instances originaires d'un concept (à ne pas confondre avec les individus OWL)
• ***f*** est une mesure de fréquence et / ou de *rank*

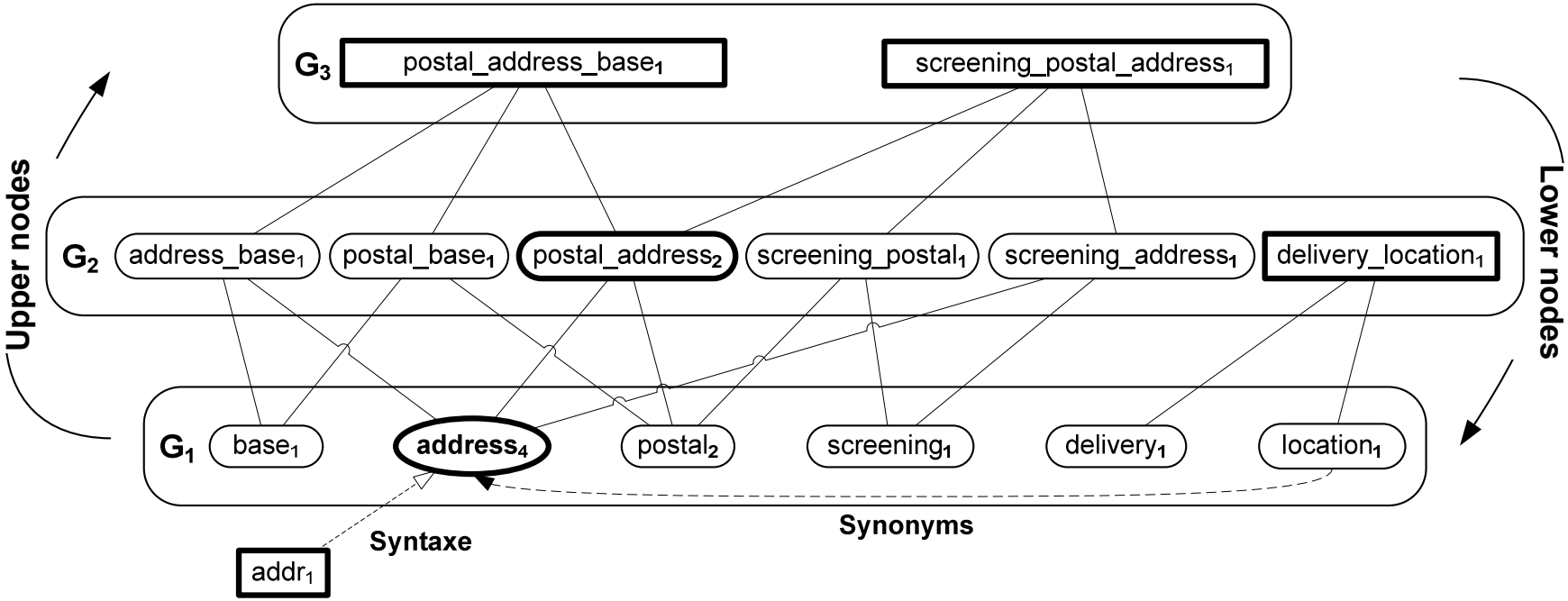
SDMO : les relations

- Types de relations entre les concepts
  - Sémantiques
  - Structurales
  - Syntaxiques (linguistiques)
  - Sources



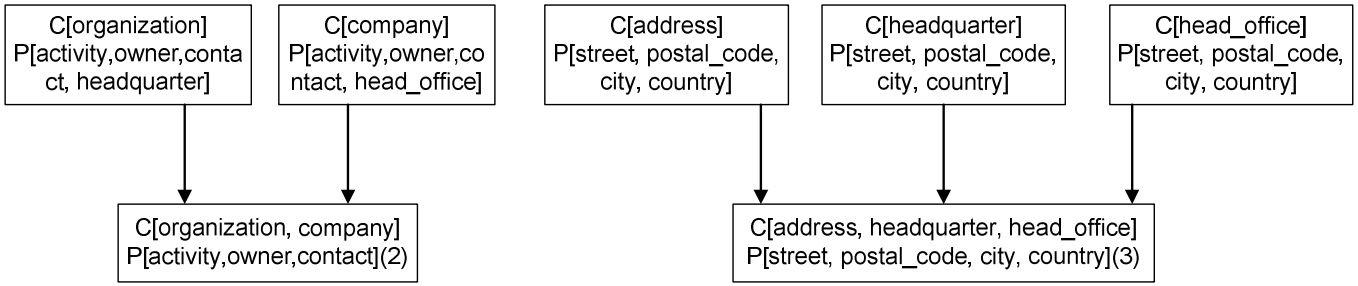
SDMO : relations sémantiques

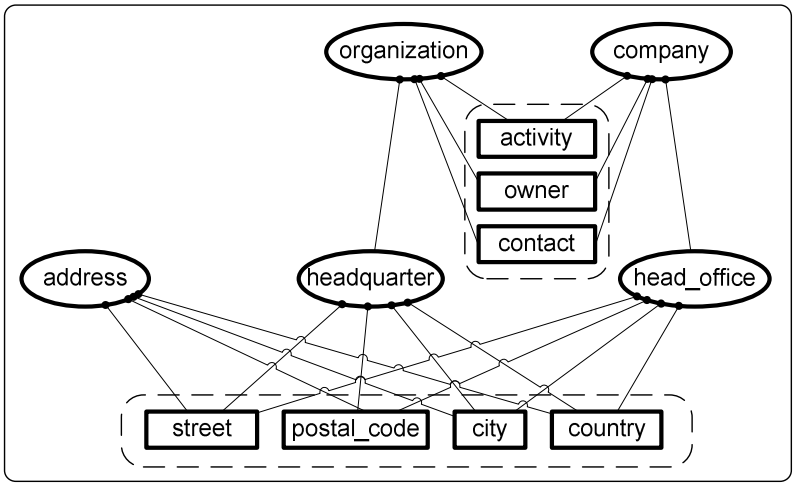
- Un système basé sur des treillis de Galois et la mesure de la fréquence
- Détection des affinités sémantiques entre les noms des concepts



SDMO : treillis des propriétés

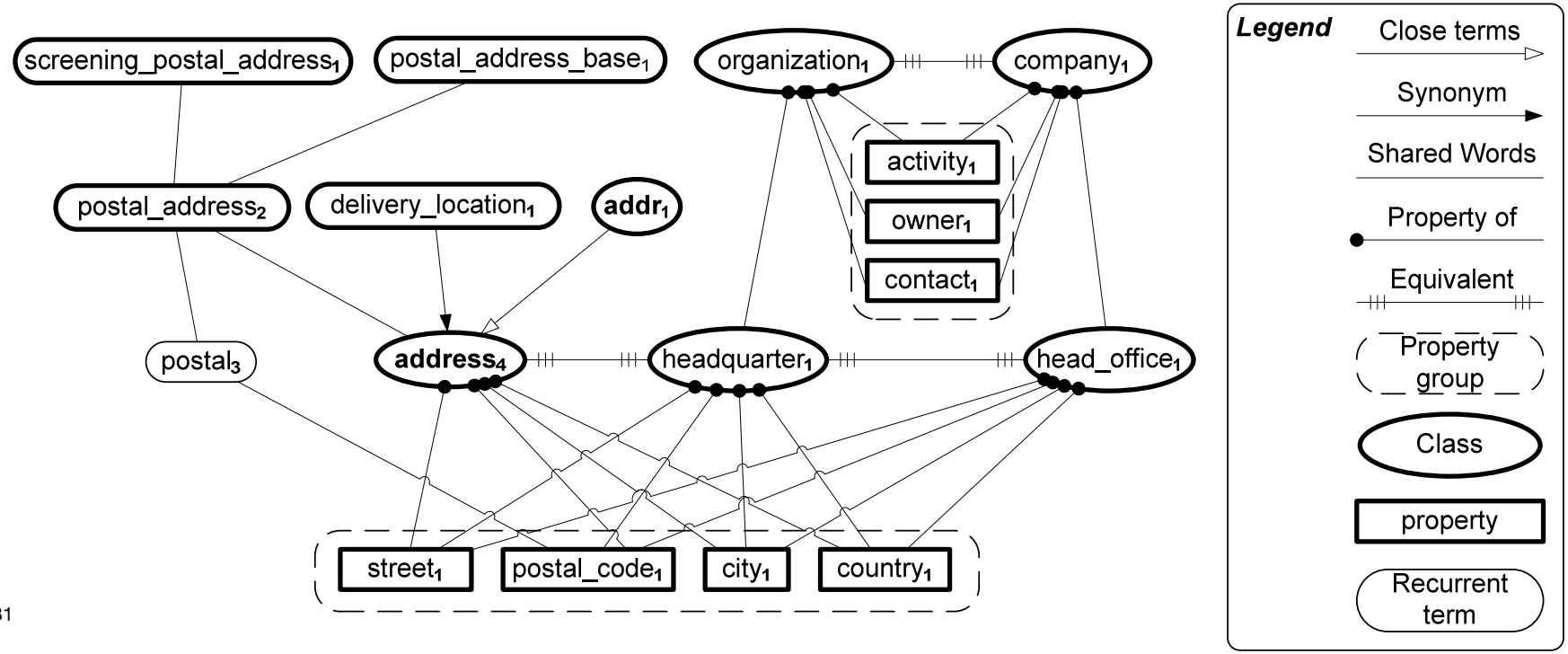
- Relations structurelles sont basées sur un treillis de Galois où les *extents* sont représentés par les concepts et les *intents* par les propriétés.





SDMO : réseau de similarité

- Mélange des relations structurelles, morphologiques et sémantiques
- Moyen pratique pour stocker et maintenir une information véridique aussi concise que possible.
- Les correspondances sont générales, valides et indépendantes du contexte d'utilisation spécifique.



2.3

XML Mining : extraction de la connaissance à partir des schémas XML

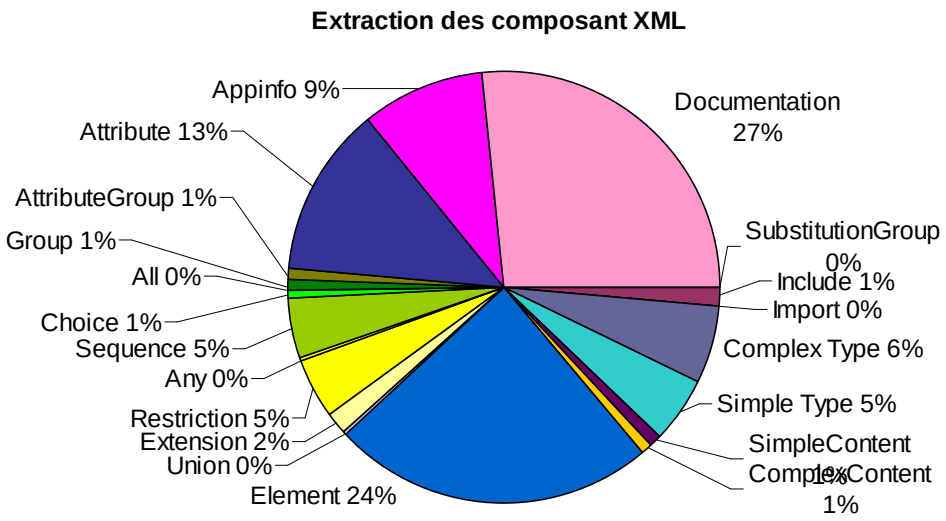
XML Mining

- Fournit l'extraction de la connaissance nécessaire pour générer l'ontologie (concepts, propriétés et relations).

- Techniques appliquées:
  - NLP - morphologique et analyse lexicale
  - Association-mining - calcul des fréquences des termes (TF) et des règles d'association
  - Sémantique - principalement pour détecter synonymie et possible homonymie
  - Treillis de Galois - regroupement sémantique et structurel de concepts similaires

XML Mining : analyse des balises

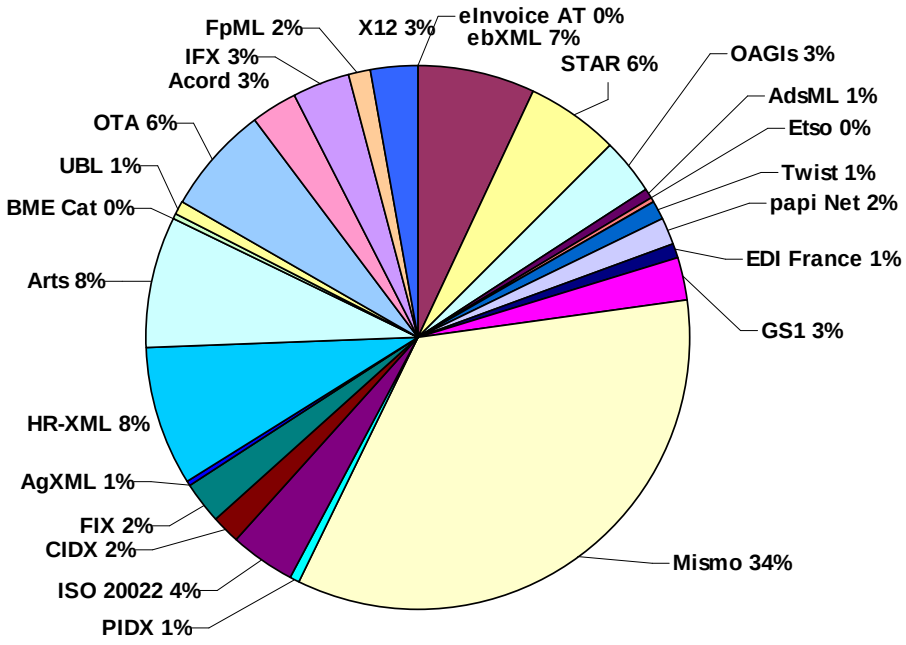
- Corpus B2B composé de :
  - 25 standards
  - 3432 fichiers XSD
  - +586.000 composants XML
  - +170.000 nommées
- 19 composants XML considérés

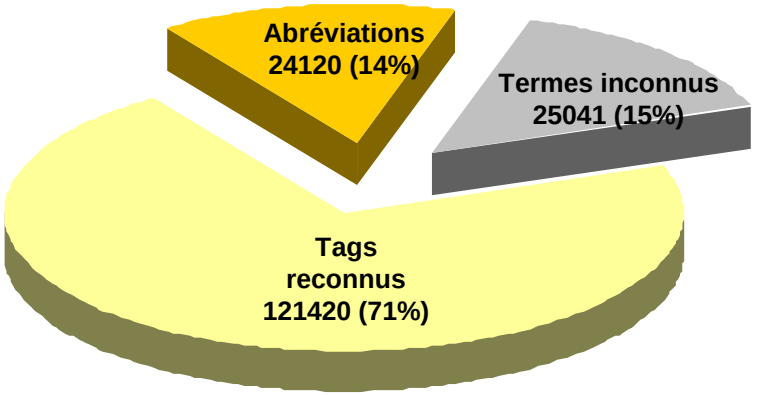


[XSD construct]	XML2OWL	OWLMAP	LDM	Janus
All	✓	✓	✓	✓
Annotation			✓	
Any			✓	✓
Appinfo				
Attribute		✓	✓	✓
AttributeGroup		✓	✓	✓
Choice	✓	✓	✓	✓
Complexcontent				✓
ComplexType	✓	✓	✓	✓
Documentation				
Element	✓	✓	✓	✓
Extension		✓	✓	✓
Group		✓	✓	✓
Import			✓	✓
Include			✓	✓
Restriction		✓	✓	✓
Sequence	✓	✓	✓	✓
SimpleContent				✓
SimpleType	✓	✓	✓	✓
SubstitutionGroup		✓		✓
Union			✓	✓
List			✓	
Min/Max Occurs	✓	✓	✓	✓
Namespace		✓		

XML Mining : quelques figures de termes extraites

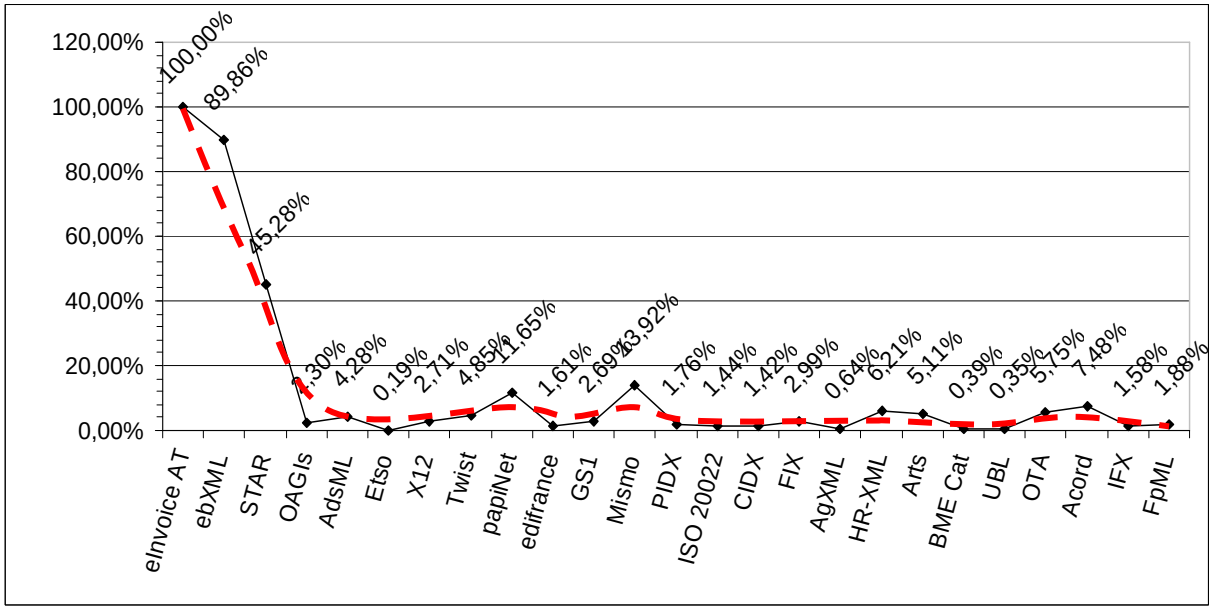
- Le 85% des termes composant les tags XML est reconnu et utilisé pour générer un premier vocabulaire contrôlé du domaine et correspond à seulement ~3000 mots





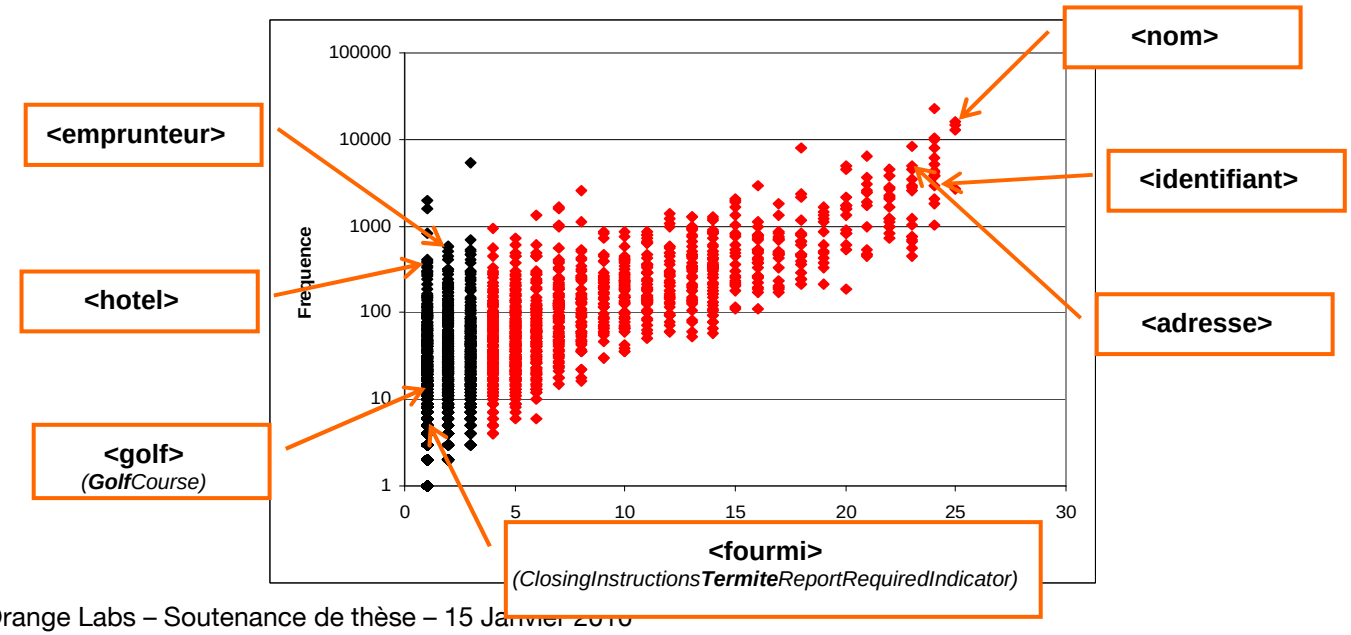
XML Mining : génération d'un vocabulaire de domaine

- Déduction d'un vocabulaire commun de domaine
- L'incrément du nombre de mots réduit considérablement avec l'ajout de nouveaux standards



XML Mining : représentativité des termes

- 170.000 balises XML ⇔ 3342 mots
- +90 % des occurrences sont présentes dans au moins 4 standards
- 2% des mots (~ 60) couvrent 40% des occurrences totales
- 40% des mots (~ 1400) sont utilisés par une seule norme
- Les mots présents dans un seul standard sont spécifiques un secteur d'activité (*e.g. hôtel, voyage, voiture, fumeur, policier, scanner, hygiénique, moléculaire…*)

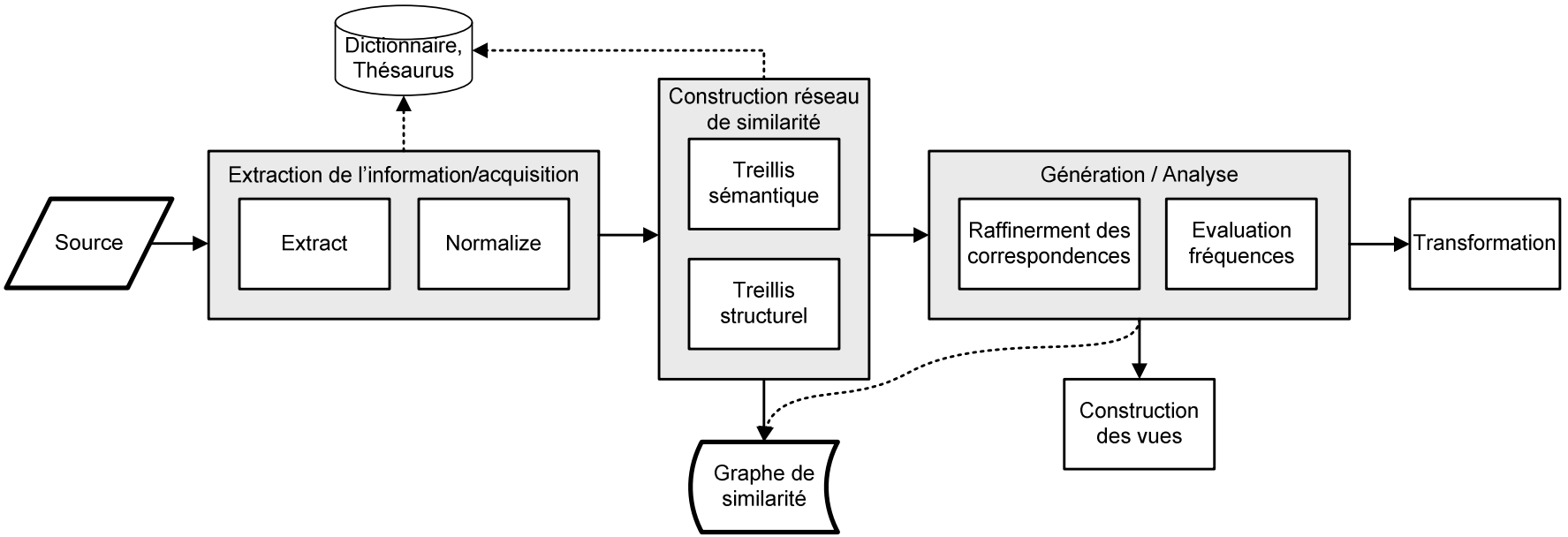


2.4

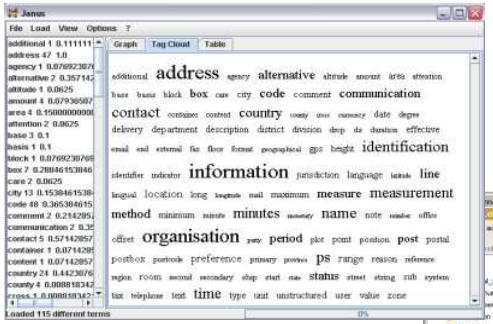
Janus : automation de la génération d'ontologie

Janus : architecture globale

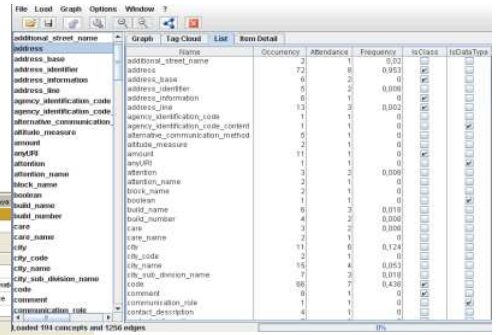
- Janus : prototype Java permettant la génération automatique d'un premier squelette d'ontologie à partir de sources XSD
- Met en œuvre une approche de *matching* complexe
- Réalise les étapes d'Extraction, Analyse, Génération et Evolution



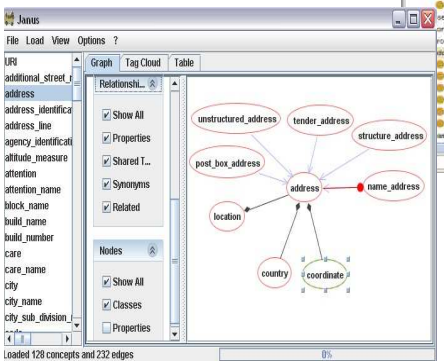
Janus : vues générées



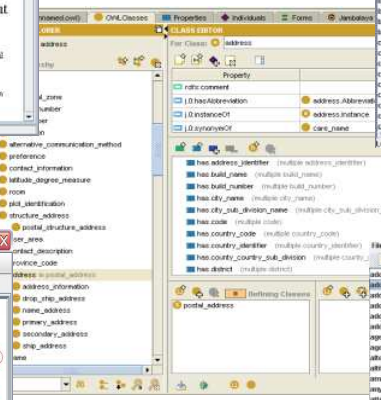
• Tag Cloud View



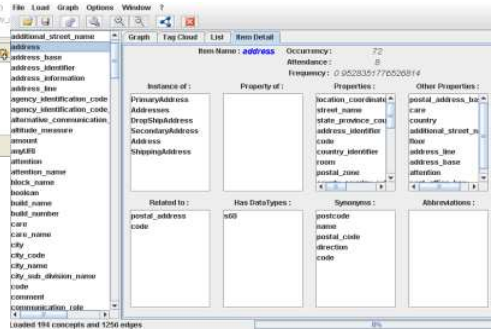
• List View



• Graphical View



• Ontology



• Concept Detail View

2.5

Evaluations

Evaluations : le corpus B2B

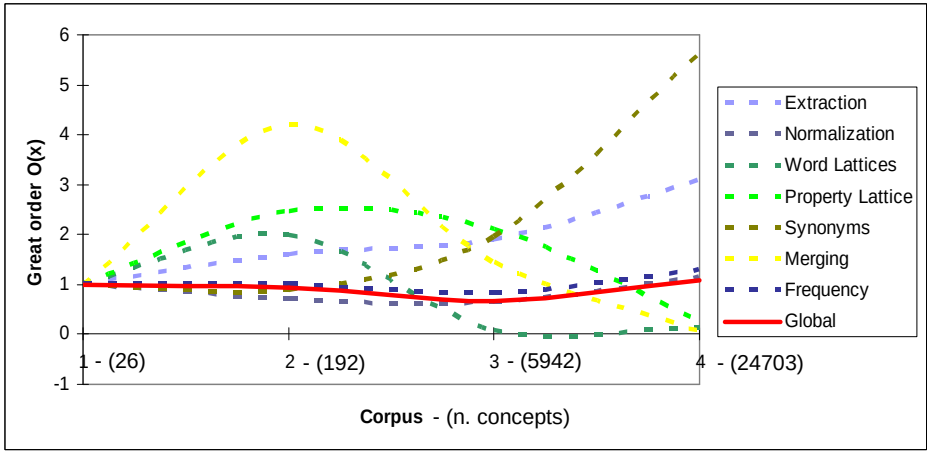
- 4 corpus (*Complete B2B* ⊃ *Invoice* ⊃ *Address* ⊃ *Coordinate)*
- 2 alignement de référence (*Coordinate, Address*)

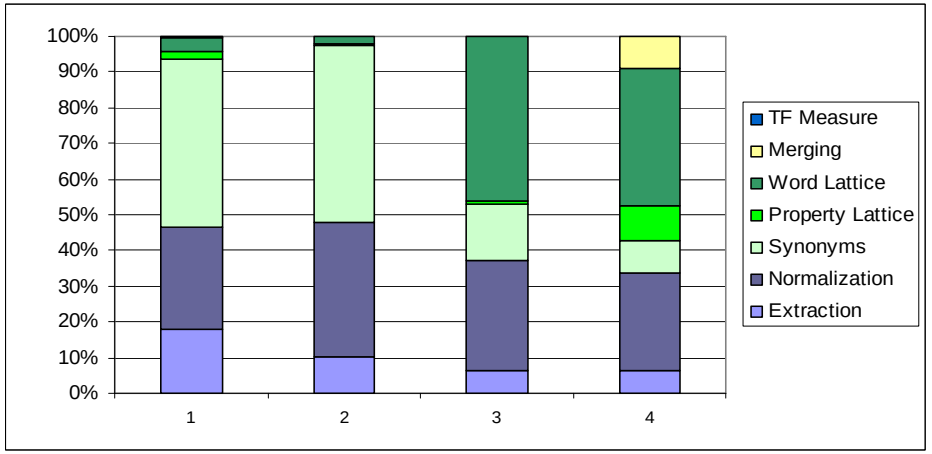
Test corpus name	Groups	Files	Extracted XSD Components	Resulting concepts
Coordinate	7	7	94	26
Address	10	15	463	192
Invoice	9	196	10002	5942*
Complete B2B	25	3432	69270	24703*

Temps d'exécution : vitesse et extensibilité

- Le système est rapide et scalable
- Améliorations possibles sur la création du treillis des mots

Unit of measure [msec]	**Information Extraction phase**		**Similarity Network construction/integration**			**Analysis computation**		
Corpus	**Extraction**	**Normalization**	**WL**	**PL**	**Synonyms**	**Merging**	**Freq.**	**Total**
Coordinate	406	656	94	47	1062	5,486	0,171	2312
Address	1251	4546	235	94	6015	6,444	0,834	12219
Invoice	22843	109813	165093	2375	57595	406	22	371797
Complete B2B	97374	423532	591600	146000	138590	138984	96,389	1561125

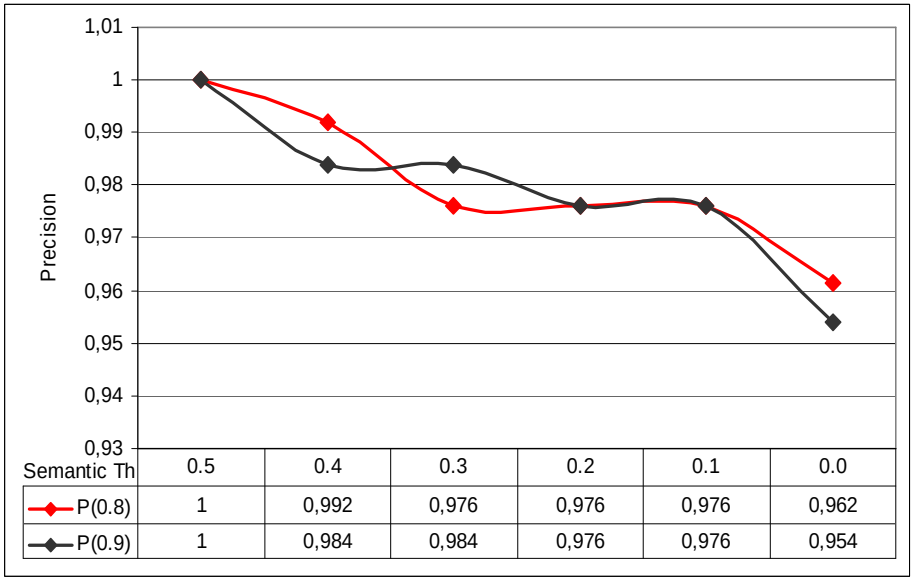


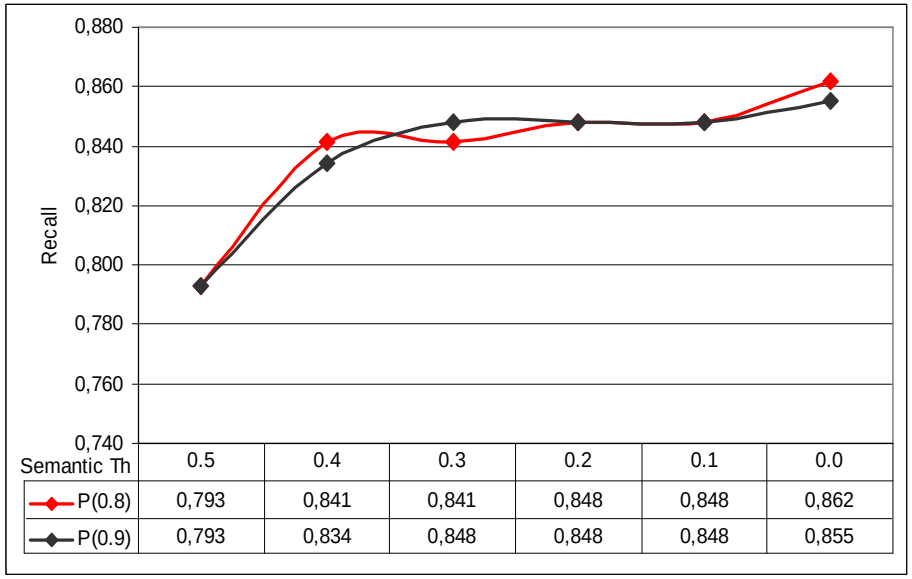


Mesure de la qualité : précision et rappel

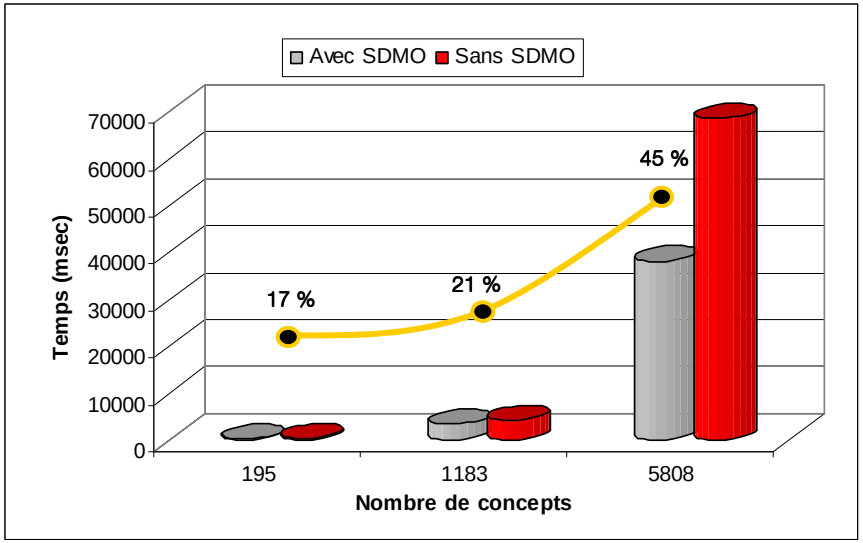
Address concepts: 192 -	Correct correspondences to provide 145					
High threshold	0.8	0.8	0.8	0.8	0.8	0.8
Low threshold	0.5	0.4	0.3	0.2	0.1	0.0
Resulting Concepts	222	214	212	211	211	207
Mergings done	115	123	125	126	126	130
Correct	115	122	122	123	123	125
Precision	**1**	**0,992**	**0,976**	**0,976**	**0,976**	**0,962**
Missing	30	23	21	20	20	20
Recall	**0,793**	**0,841**	**0,841**	**0,848**	**0,848**	**0,862**

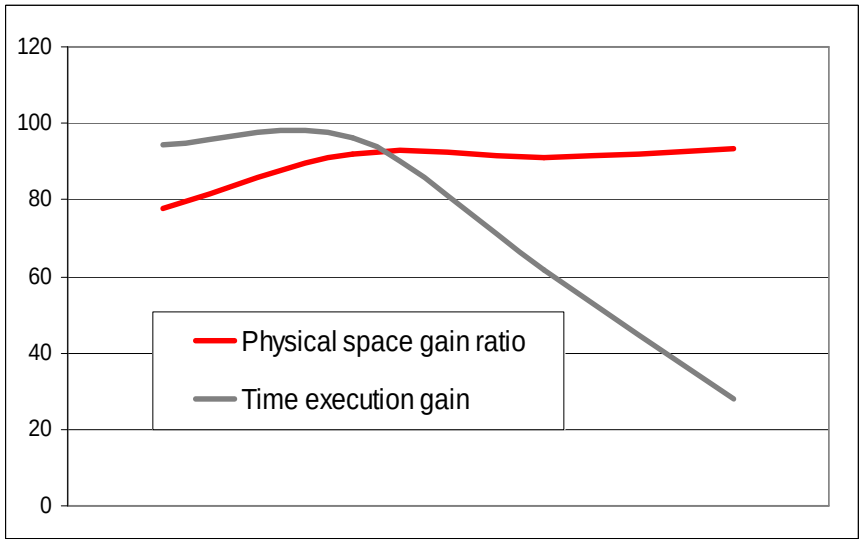
Address concepts: 192 -	Correct correspondences to provide 145					
High threshold	0.9	0.9	0.9	0.9	0.9	0.9
Low threshold	0.5	0.4	0.3	0.2	0.1	0.0
Resulting Concepts	222	214	212	211	211	207
Mergings done	115	123	125	126	126	130
Correct	115	121	123	123	123	124
Precision	**1**	**0,984**	**0,984**	**0,976**	**0,976**	**0,954**
Missing	30	24	22	22	22	19
Recall	**0,793**	**0,834**	**0,848**	**0,848**	**0,848**	**0,855**





Gain de temps et format de stockage





- **Gain de temps** entre un système avec SDMO et un système traditionnel **augmente** avec la taille et le nombre des sources

- Janus offre un **format de stockage** pour le modèle.
  - **Améliore** encore le temps d'exécution
  - **Réduit l'espace** de stockage nécessaire aux sources

3

Conclusion et perspectives

Perspectives

- Janus sous forme de **modules réutilisable** (API) par d'autres systèmes de *matching*
- Janus sous forme de **plug-in** dans des outils UML et XML :
  - Générer automatiquement des ontologies d'entreprise
  - Plus simplement pour transformer les Schémas XML en OWL
- Système de **recommandation** sur le choix d'un standard à utiliser dans une transaction d'affaire
- **Enrichissement dynamique** d'ontologie d'un catalogue de services (use case projet EU SERVERY …en cours)

Travaux futurs

- Formalisation détaillée de la **méthodologie**
- Automatisation de l'intégration de **sources hétérogènes** (structurés et pas structurés)
- Intégration de systèmes de **raisonnement**

Synthèse

- Définition du **cycle de vie** de la génération automatique d'ontologie
- Un système avancé d'**extraction** de la connaissance à partir des **schémas XML** multiples
- Un **modèle sémantique** qui permet de réduire le temps de calcul des systèmes de *matching*
- Un système complexe de **génération/déduction** de connaissance ontologique exprimé en OWL

merci

orange™

Bibliographie Personnelle

International Conferences

- Mathieu Boussard, Vincent Hiribarren, Jean Pierre Le Rouzic, Stéphanie Fodor, Ivan Bedini, Noel Crespi, Gabor Marton, David Moro, Manuel Macias, Oscar Lorenzo Dueñas, Benjamin Molina. Servery: Web Telco Marketplace. Information and Communication Technologies –Mobile Summit 2009. 10 - 12 June 2009, Santander, Spain.
- Ivan Bedini, Benjamin Nguyen and Georges Gardarin. B2B Automatic Taxonomy Construction. In Proceedings 10th International Conference on Enterprise Information Systems. 12 - 16, June 2008 Barcelona, Spain.

International Workshops

- Jérôme Le Moulec, Jacques Madelaine and Ivan Bedini. Discovery Services Interconnection. International Workshop on RFID Technology. Mai 2009, Milan, Italy.

International Demos

- Ivan Bedini, Benjamin Nguyen and Georges Gardarin. Janus: Automatic Ontology Construction Tool. Demo-Poster Session. 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns. September 2008, Acitrezza, Italy.
- Ivan Bedini, Benjamin Nguyen and Georges Gardarin. Janus: Automatic Ontology Builder from XSD files. Developer track at 17th International World Wide Web Conference (WWW2008). Beijing, China, April 21 - 25, 2008

French Conferences and Demos

- Ivan Bedini, Georges Gardarin, Benjamin Nguyen. Deriving Ontologies from XML Schema. In Proceedings of the Entrepôts de Données et Analyse en Ligne (EDA), France, June 2008. RNTI, Vol. B-4, 3-17 (Invited Paper).
- Ivan Bedini, Fabrice Bourge, Benjamin Nguyen. RepXML: Experimenting an ebXML Registry to Store Semantics and Content of Business Messages. Developer Track at BDA 2006. Lille, France. October 2006.

Submitted

- Ivan Bedini, Georges Gardarin, Benjamin Nguyen. Semantic Web and e-business. Submitted Chapter for the book « Electronic Business Interoperability: Concepts, Opportunities and Challenges », IGI Global publisher. December 2009.
- Emmanuel Bertin, Ivan Bedini, Nassim Laga, Benoit Cristophe, Benjamin Molina. Selecting the best available service at runtime: the concept of abstract services. Submitted to IEEE transactions on Software engineering journal. November 2009.

Patents

- Ivan Bedini, Emmanuel Bertin, Nassim Laga. Dynamic selection of the best web service meeting user requirements process. Patent INPI number: 0954427 - 06/2009 *(Pending)*
- Ivan Bedini. Procedure for the automation of data sources matching combining semantic and structural properties. Request for patent INPI number: 08 58363 – 12/2008 *(Pending)*

Standardisation Activities, main contributions

- Fabrice Bourge, Ivan Bedini. UN/CEFACT Registry Implementation Specification. UN/CEFACT ICG Standard Draft. 2007
- ebXML Registry Profile for OWL-Lite. OASIS Standard Technical Note (Contributor). 2005
- OASIS/ebXML Registry Information Model Specification V3.0. OASIS Standard Specification (Also ISO 15000, part 3 and 4 Standard) (Contributor). 2005
- OASIS/ebXML Registry Services and Protocols v3.0. OASIS Standard Specification (Also ISO 15000, part 3 and 4 Standard) (Contributor)
- Ivan Bedini, Fabrice Bourge, Francis Berthomieu, Fabrice Jeanne, Sébastien Wafflart. EDIFRANCE RepXML Project Overview. UN/CEFACT ICG Deliverable. 2005.