

Deriving Ontologies from XML schemas

Georges Gardarin*

Ivan Bedini**, Benjamin Nguyen*

* PRiSM, UVSQ, Versailles

** Orange Labs, Caen



Plan

1. Introduction et bases
2. Ontologies pour EDA
3. Construction d'ontologies
4. Janus : de xsd à owl
5. Conclusion

1. Introduction

- ◆ Contexte : **Entrepôts de Données et Analyse en ligne**
 - Datawarehouse, OLAP, Datamining
 - Les ontologies paraissent un outil fort utile ...
 - À la mode avec le Web sémantique
- ◆ Exemples d'applications :
 - OLAP
 - Intégration de données et services
 - Navigation en analyse de données
 - Data Mining
 - Fouille de texte, de pages Web, ...
 - Web sémantique et annotations



Ontologie : définition informelle

◆ Wikipedia

- In philosophy, ontology is the most fundamental branch of metaphysics. It studies being or existence and their basic categories and relationships, to determine what entities and what types of entities exist. Ontology thus has strong implications for conceptions of reality.

Plus formellement

- ◆ Un ensemble de classes (concepts)
 - $\{C1, C2, \dots C_m\}$
- ◆ Un ensemble de propriétés (facettes)
 - Propriétés valeurs (attributs) : $\{A_{ij}\}$
 - Propriétés objets (relations) : $\{R_{ij}\}$
- ◆ Un ensemble de relations de généralisation
 - $\{ISA_i\}$; chacune définit une hiérarchie entre classes
 - Sémantique d'inclusion; ordre partiel sur les classes
- ◆ Des instances de classes (objets)
- ◆ Des règles de déduction entre classes ou propriétés

Exemple: Wine ontology (Abstract Syntax)

- ◆ **Definition of classes**

Class(w:Liquid partial owl:Thing)
Class(w:Person partial owl:Thing)
Class(w:Wine partial w:Liquid)
Class(w:Coca partial w:Liquid)
Class(w:Drinker partial w:Person)

- ◆ **Constraints on classes**

DisjointClasses(w:Coca w:Wine)

- ◆ **Definition of object properties**

ObjectProperty(w:Drink
domain(w:Person)
range(w:Liquid))
ObjectProperty(w:Owner
Functional domain(w:Wine)
range(w:Person)

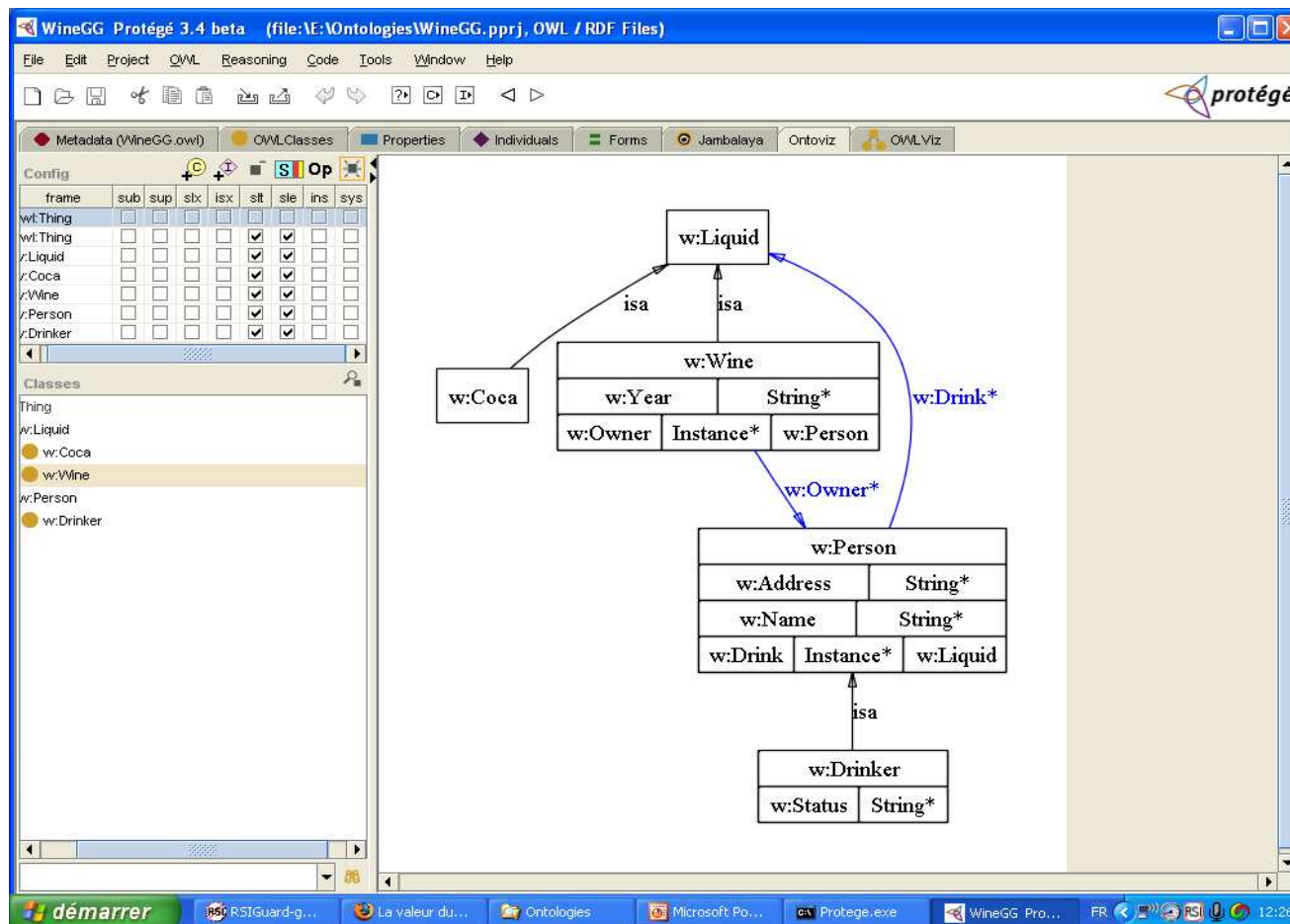
- ◆ **Definition of data properties**

DatatypeProperty(w:Name
Functional domain(w:Person)
range(xsd:string))
DatatypeProperty(w:Status
domain(w:Drinker)
range(xsd:string))
DatatypeProperty(w:Address
Functional domain(w:Person)
range(xsd:string))
DatatypeProperty(w:Year
domain(w:Wine) range(xsd:gYear))
DatatypeProperty(w:Millage)
EquivalentProperties(w:Millage
w:Year)

Instances

- ◆ Individual(w:Coca_3 annotation(rdfs:comment "Not very tasty !" xsd:string) type(w:Coca))
- ◆ Individual(w:Person_4 type(w:Person) value(w:Drink w:Wine_1) value(w:Drink w:Wine_2) value(w:Address "6 Wine Street Paris 75015" <xsd:string>)) value(w:Name "Gardarin" <xsd:string>))
- ◆ Individual(w:Wine_1 type(w:Wine) value(w:Wineyard "Beaujolais" <xsd:string>) value(w:Year "2000" xsd:gYear>))

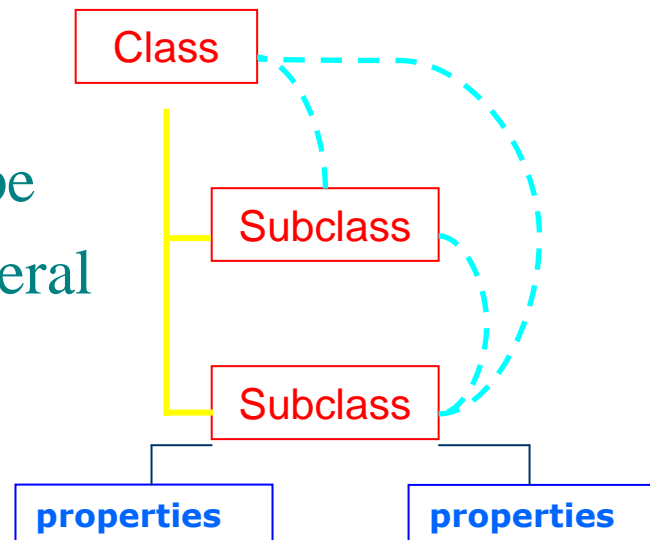
Vue graphique



RDFS: décrire un modèle objet

◆ Classes

- `rdfs:Resource`
- `rdfs:Class`
- `rdfs:Literal`
- `rdfs:Datatype`
- `rdf:XMLLiteral`
- `rdf:Property`



◆ Properties

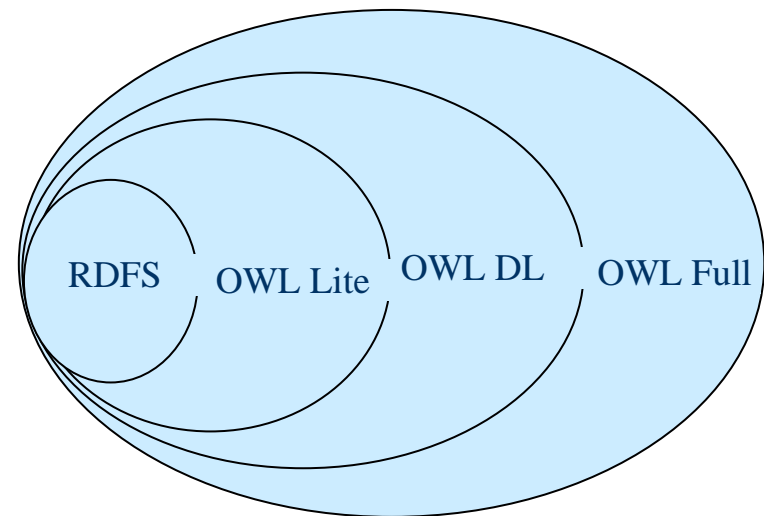
- `rdfs:range`
- `rdfs:domain`
- `rdf:type`
- `rdfs:subClassOf`
- `rdfs:subPropertyOf`
- `rdfs:label`
- `rdfs:comment`

OWL

- ◆ Offre un langage standard pour définir des ontologies
- ◆ Extension de RDFS
- ◆ Etend les constructions de base pour améliorer :
 - L'interopérabilité (e.g., equivalences)
 - Le raisonnement (e.g., description logic)
 - Les évolutions (e.g., integration, version)
- ◆ Raisonnement basé sur certaine Logique de Description
 - Concepts, Relations binaires, Constructeurs (\cup , \cap , \neg)
 - Concepts représentent des ensembles d'individus
 - Sémantique récursive pour inférer à partir des concepts atomiques

OWL: les niveaux

- ◆ OWL Lite :
 - Cardinalités limitées à 0 ou 1
 - Hiérarchies de classes
 - Contraintes simples
- ◆ OWL DL :
 - Logique de description
 - Dédution décidable
- ◆ OWL Full :
 - Complet
 - Non décidable



Qq constructions OWL Lite

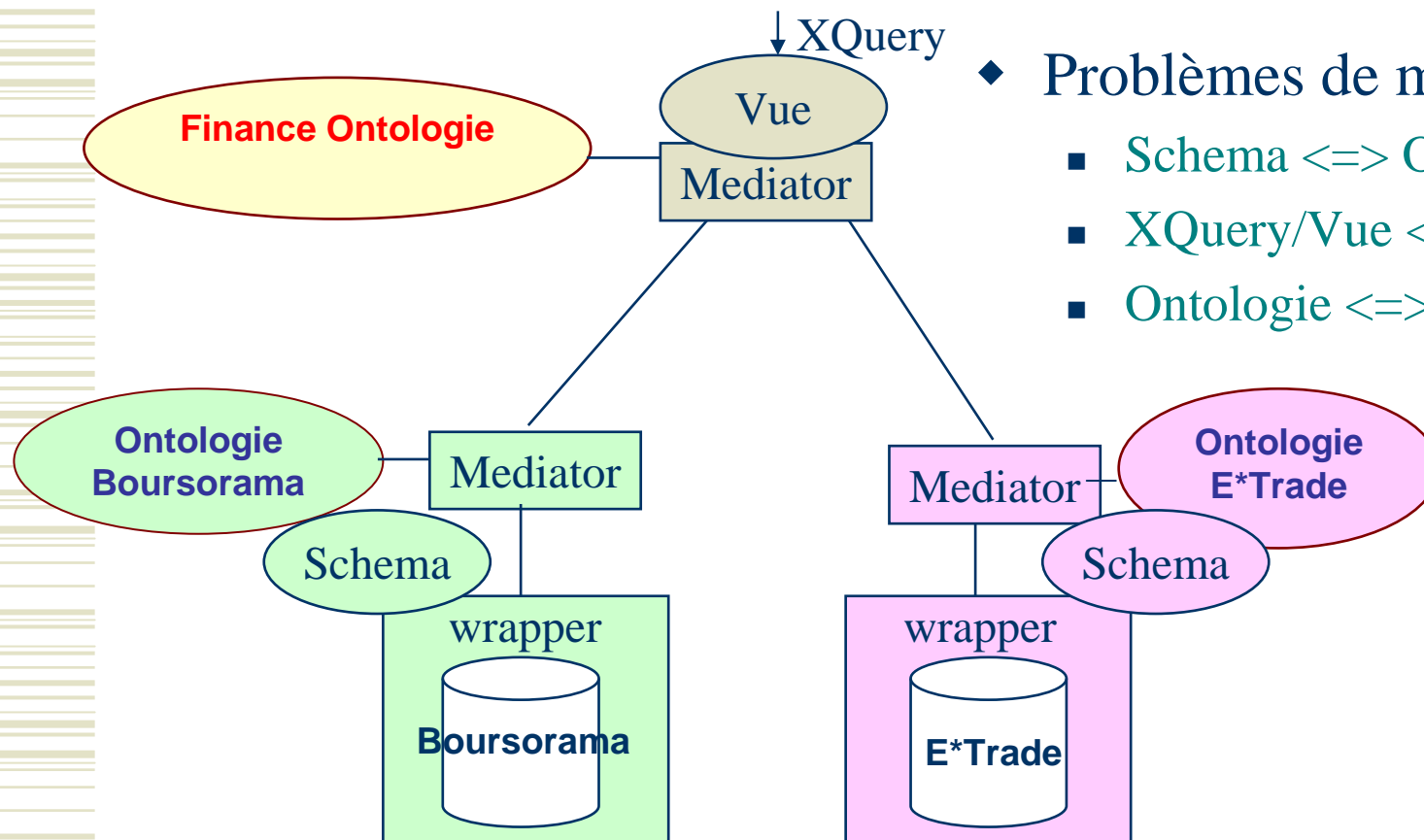
- ◆ (In)Equality:
 - equivalentClass
 - equivalentProperty
 - sameAs
 - differentFrom
 - allDifferent
- ◆ Property Characteristics:
 - inverseOf
 - TransitiveProperty
 - SymmetricProperty
 - FunctionalProperty
 - InverseFunctionalProperty
- ◆ Property Type Restrictions:
 - allValuesFrom
 - someValuesFrom
- ◆ Restricted Cardinality:
 - minCardinality (only 0 or 1)
 - maxCardinality (only 0 or 1)
 - cardinality (only 0 or 1)
- ◆ Header Information:
 - ontology
 - imports



2. Intérêt des ontologies pour EDA

- ◆ Intégration de sources hétérogènes
- ◆ Requêtes distribuées: médiation
- ◆ Réseau P2P : ubiquité, uniformisation, description
- ◆ Dimensions des cubes: zooms et détails

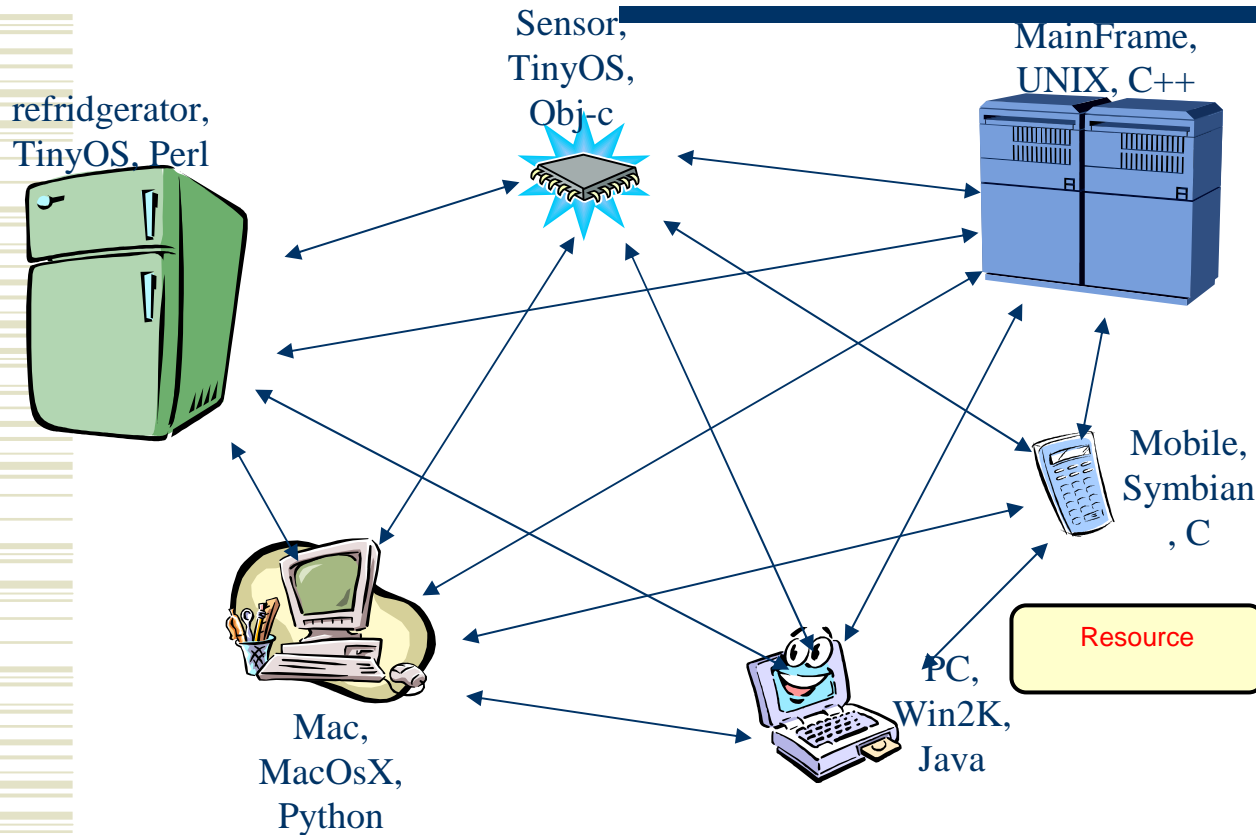
Médiation de données hétérogènes



◆ Problèmes de mappings :

- Schema \Leftrightarrow Ontologie
- XQuery/Vue \Leftrightarrow Ontologie
- Ontologie \Leftrightarrow Ontologie

Réseau P2P Ubiquitaire

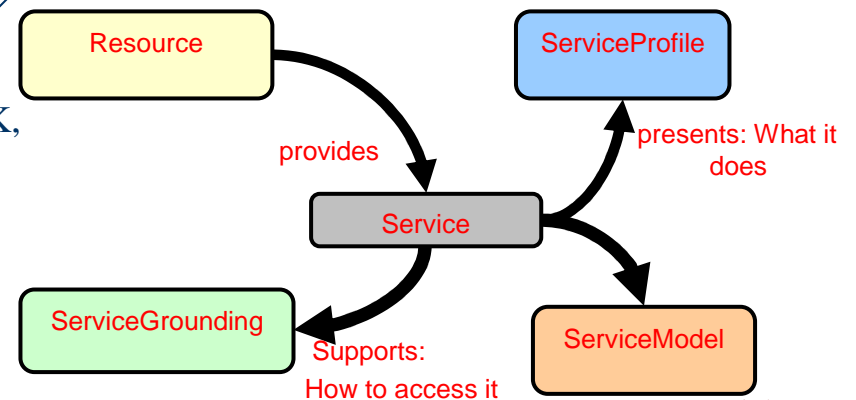


◆ Problème :

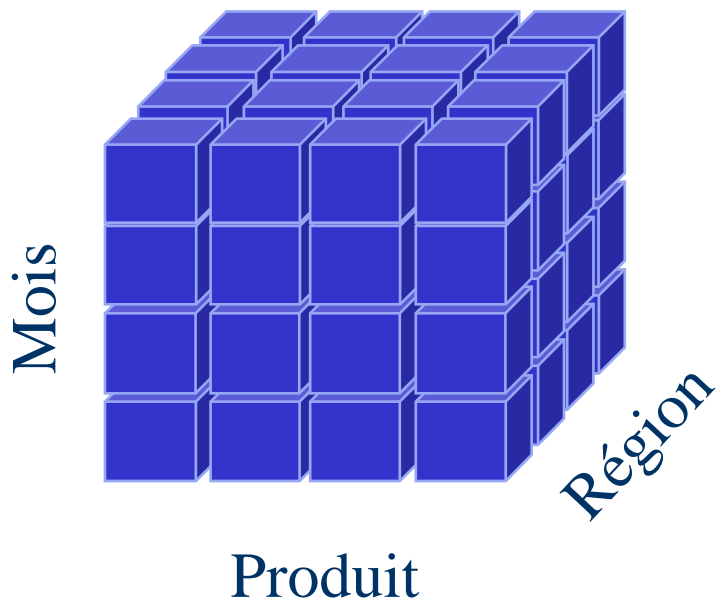
- Langage
- Interfaces

◆ Approche :

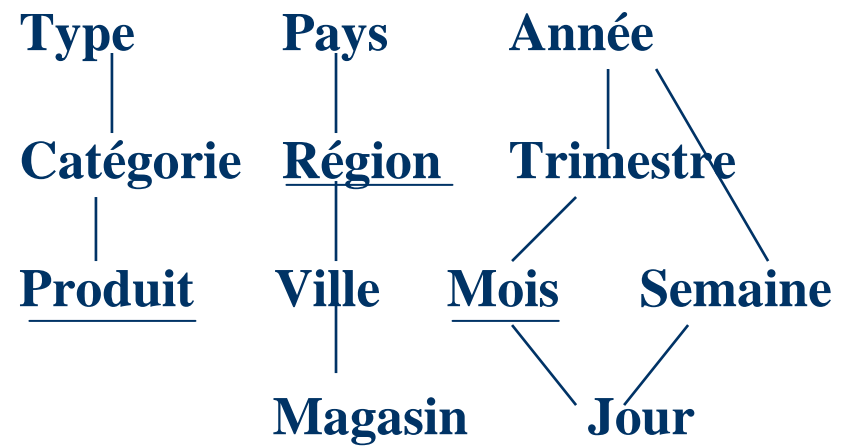
- Ontologie de services et profils
- OWL-S ?



Cube de données



Ontologie



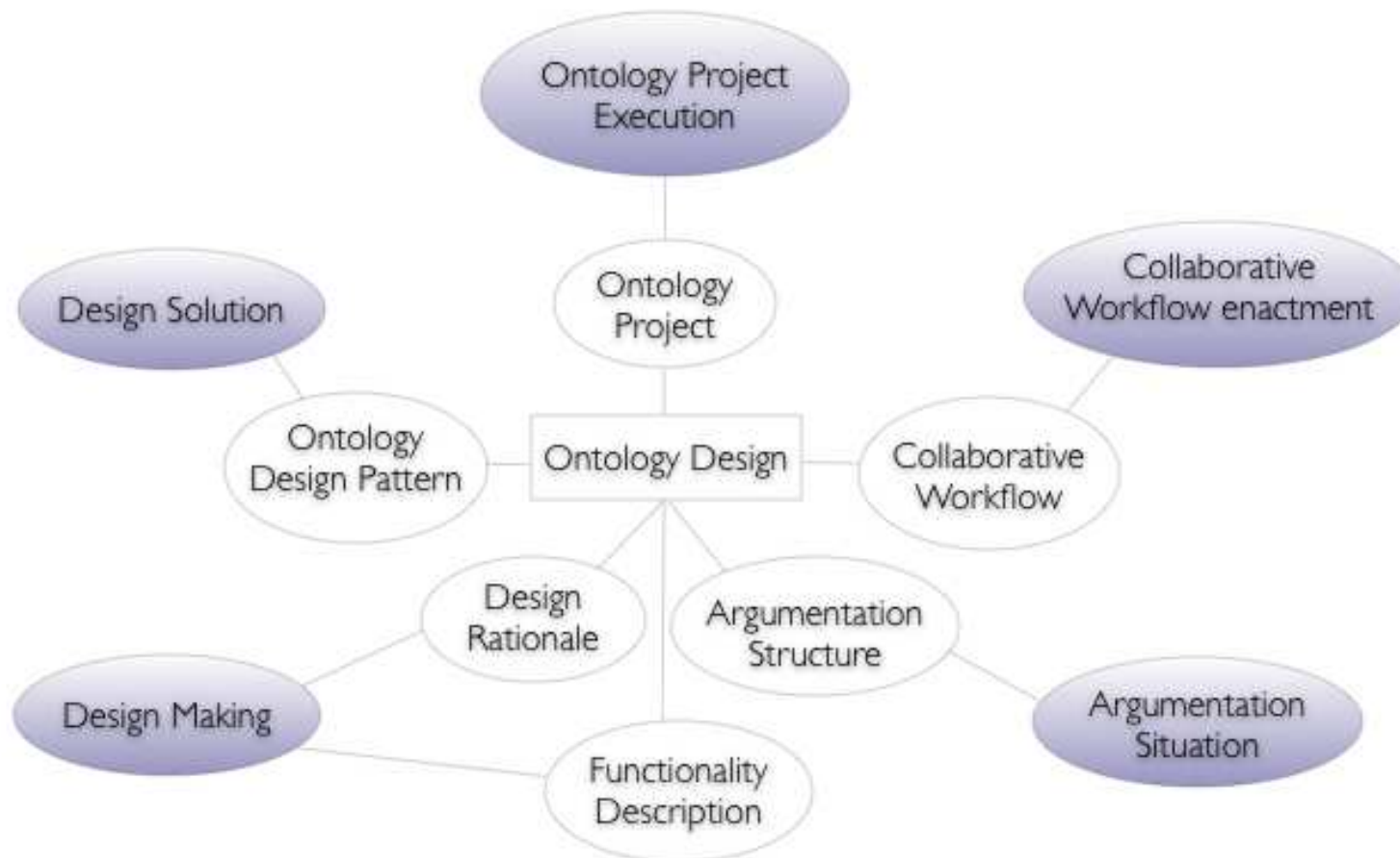
3. Construire une ontologie

- ◆ Mémorisation
 - Vue complète des concepts, propriétés et règles
 - Identification des similarités et différences
 - Contraintes d'intégrité : e.g., cardinalités, inclusion, inverse
- ◆ Evolutivité
 - Construction progressive
 - Alignement et différenciation des nouveaux termes
 - Compréhension dynamique d'un domaine
- ◆ Polysémie
 - Support de plusieurs sens pour un mot et de même sens pour des mots différents
 - Gestion de contexte incluant propriétés et types
 - Intégration de lexiques / thésaurus grammaticaux
- ◆ Automatisation
 - Pour produire, maintenir et enrichir l'ontologie
 - Intérêt à s'appuyer sur une BD

Méthodes

- ◆ Manuelles
 - Trois syntaxes pour OWL ...
- ◆ Graphiques
 - e.g. Protégé, Semantic Work,
 - Visualisation : graphes 2D, 3D, zoom, boîtes liées, UML
- ◆ Automatisées
 - À partir de textes, dictionnaires, modèles, patterns ...
 - Fouille de pages Web, moteur de recherche, ...
 - Workflows de conception collaborative

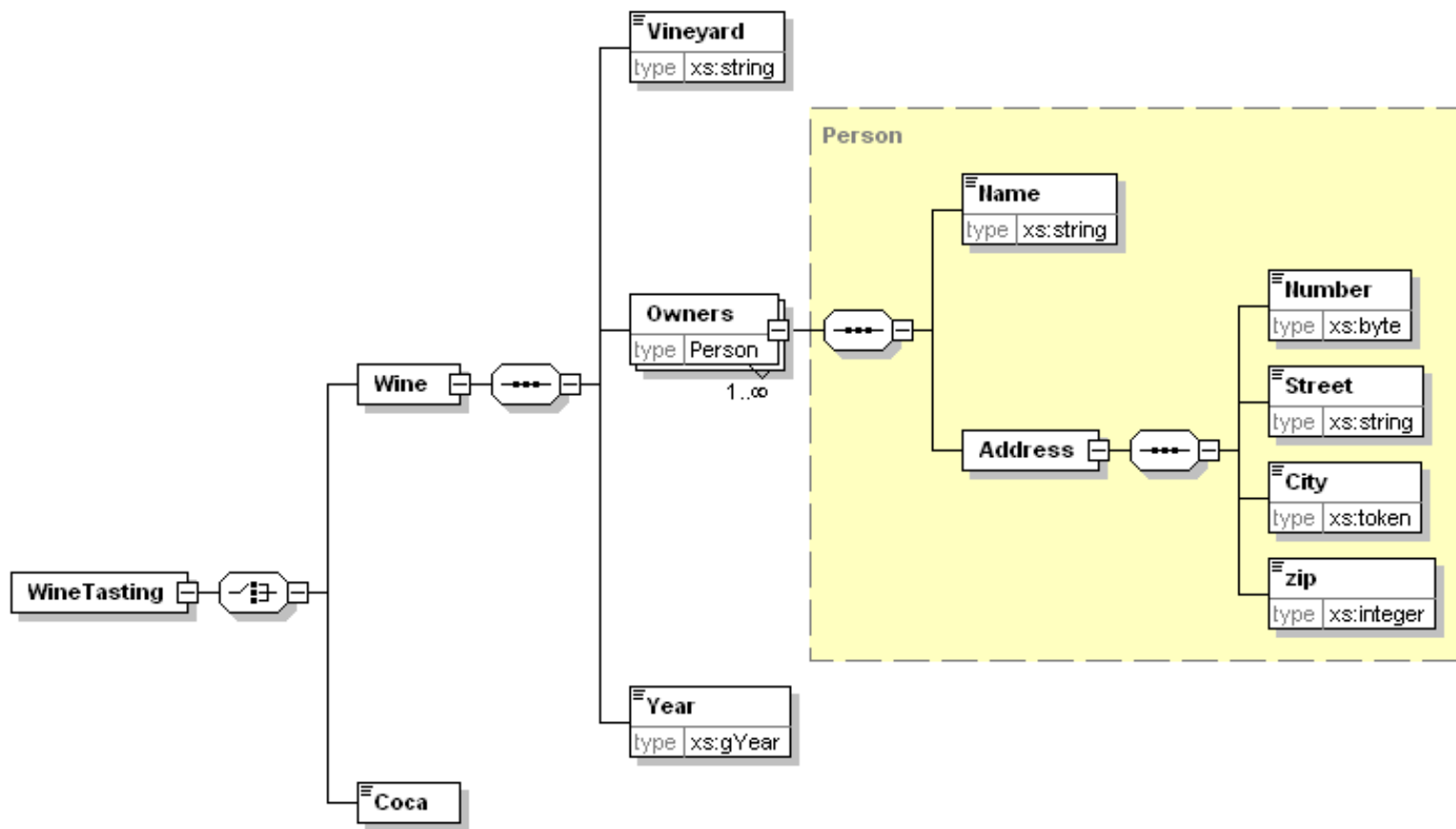
COLLABORATIVE ONTOLOGY DESIGN (CODO)



Proposition: Partir de Schémas XML

- ◆ Décrivent les arbres de structures XML
- ◆ Types simples variés
 - Propriétés valeurs (attributs ou éléments)
- ◆ Types complexes
 - Propriétés objet (imbrication confuse)
- ◆ Il existe beaucoup de schémas par domaine
 - B2B, Santé, ...
 - Précèdent la construction d'ontologie
- ◆ Des exemples, standards documentés, processus métiers, ...

Exemple



XML Schéma et Ontologie

- ◆ Récupérer plusieurs schémas
- ◆ Modélisation sémantique simple
- ◆ Compléter la sémantique des schémas
- ◆ Unifier, intégrer les schémas
- ◆ Cas du B2B: nombreux standards
 - UBL, ebXML core components, CXML, OAGIS, STAR, PapiNet, ...

4. Janus



- ◆ Outils pour la construction automatique d'ontologies à partir de fichiers XSD
- ◆ Applique des techniques de fouille de texte
 - adaptation de plusieurs techniques provenant du text-mining (lexique, vectorisation, distance, ...)
 - et de recherche / extraction d'information
- ◆ à des fichiers XML



Janus : Objectifs



- Réaliser un système capable d'acquérir des connaissances et d'ajouter à la volée des nouvelles informations à partir d'un corpus source
- Générer et maintenir une "mémoire collective", centrée sur une base de concepts, pour faciliter la découverte de classes et propriétés similaires ou non
- Limiter le plus possible l'intervention humaine dans le processus de construction d'une représentation de la connaissance pour un domaine

Janus : Etapes proposées

1. Extraction

- Recherche de connaissance et normalisation du vocabulaire

2. Analyse

- Détermination des classes, propriétés et types de données
- Construction du réseau sémantique de concepts (règles, similarités)

3. Génération

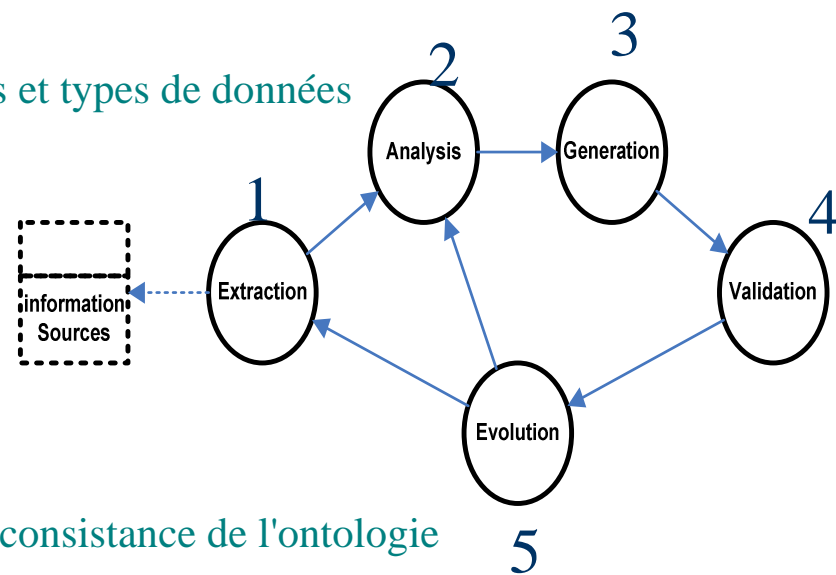
- Production d'une vue globale en intégrant les concepts similaires
- Transformation en format de sortie (OWL)

4. Validation

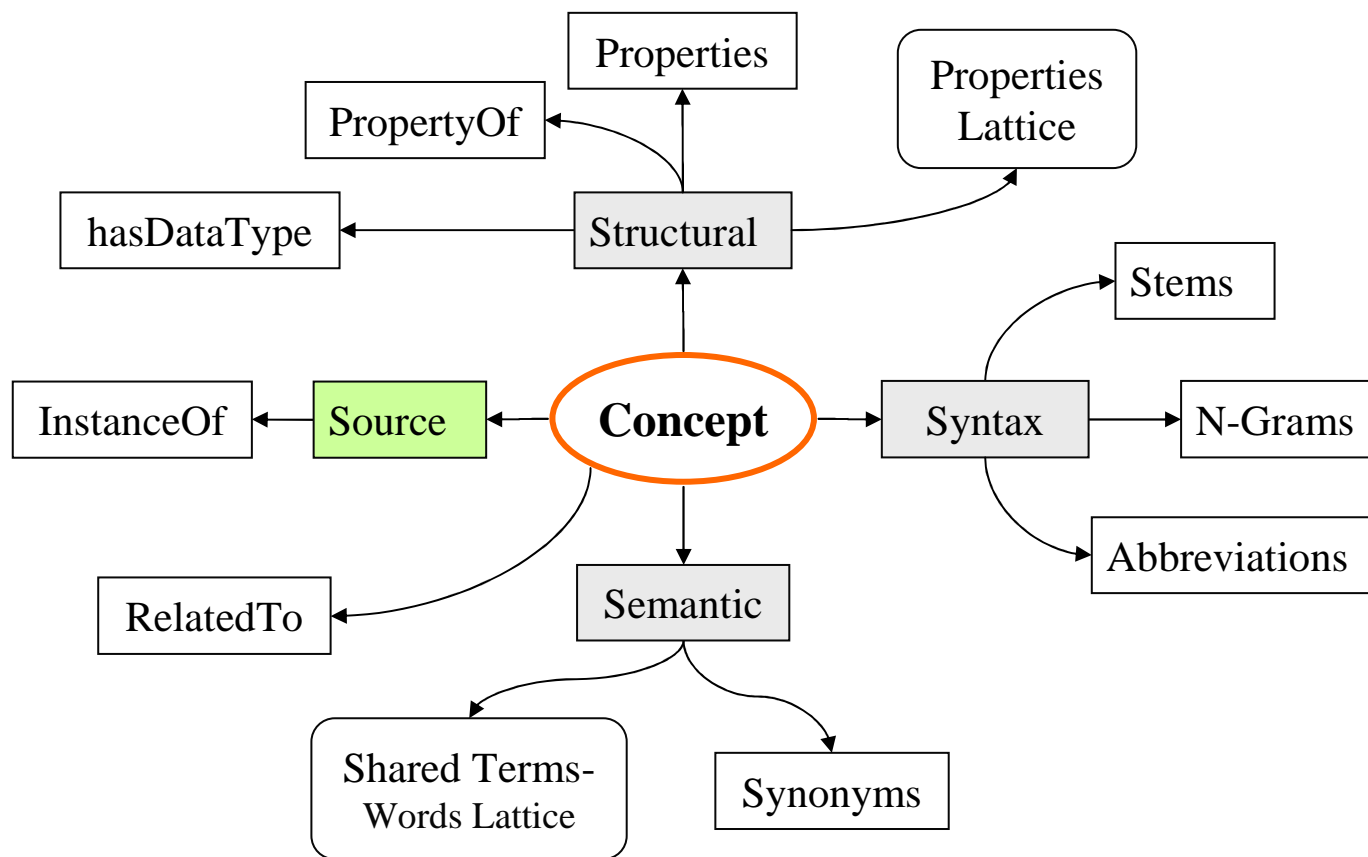
- Vérification de la cohérence et de la consistance de l'ontologie

5. Evolution

- Enrichissement de l'ontologie en ajoutant des nouvelles sources



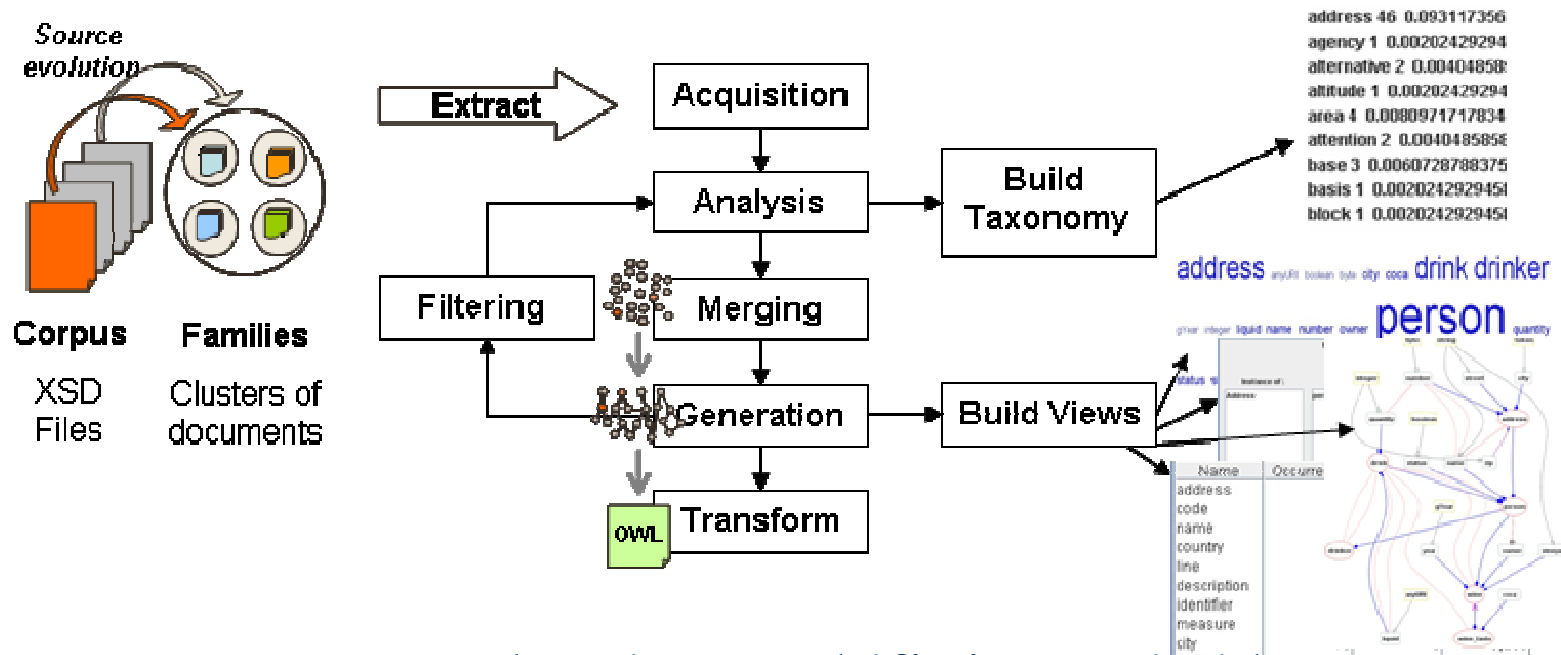
Janus : Modèle sémantique



Janus : Conceptualisation

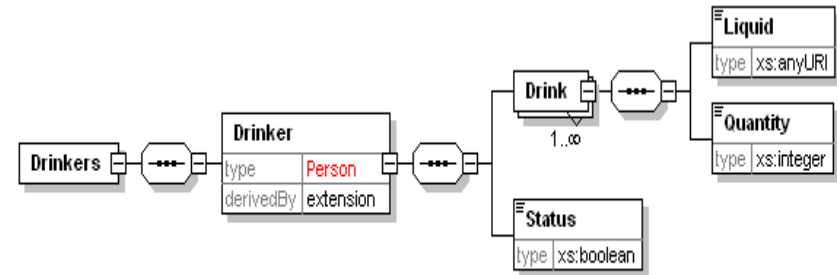
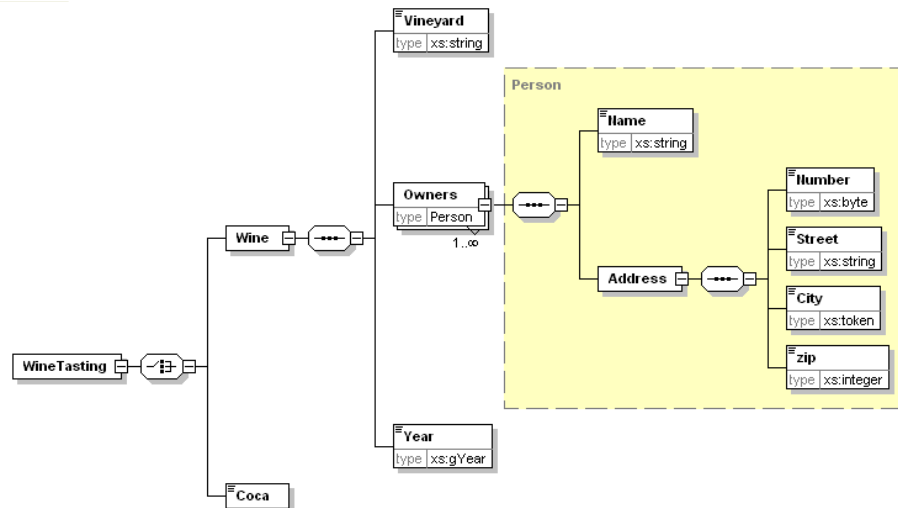
XSD Structure	Concepts
xs:complexType	Concept class
xs:complexType with declared xs:simpleContent	Concept datatype
Element with attribute "ref" to xs:complexType	Concept class with propertyOf relationship
Named xs:element with attribute "type"	Concept class with Is a relationship
Named xs:element	Concept class
xs:simpleType	Concept datatype
Attributes of xs:element or xs:complexType	Concept properties
xs:extension et xs:restriction	Datatype property and is a relationship
xs:minOccurs, xs:maxOccurs	Respective cardinalities
xs:sequence, xsd:all	Concept properties
xs:choice	Disjointness concepts

Janus : Architecture



- ◆ Met en œuvre les étapes définies précédemment :
 - Extraction, Analyse, Génération et Evolution
 - Etape de Validation en cours d'étude...

Janus : Application aux vins

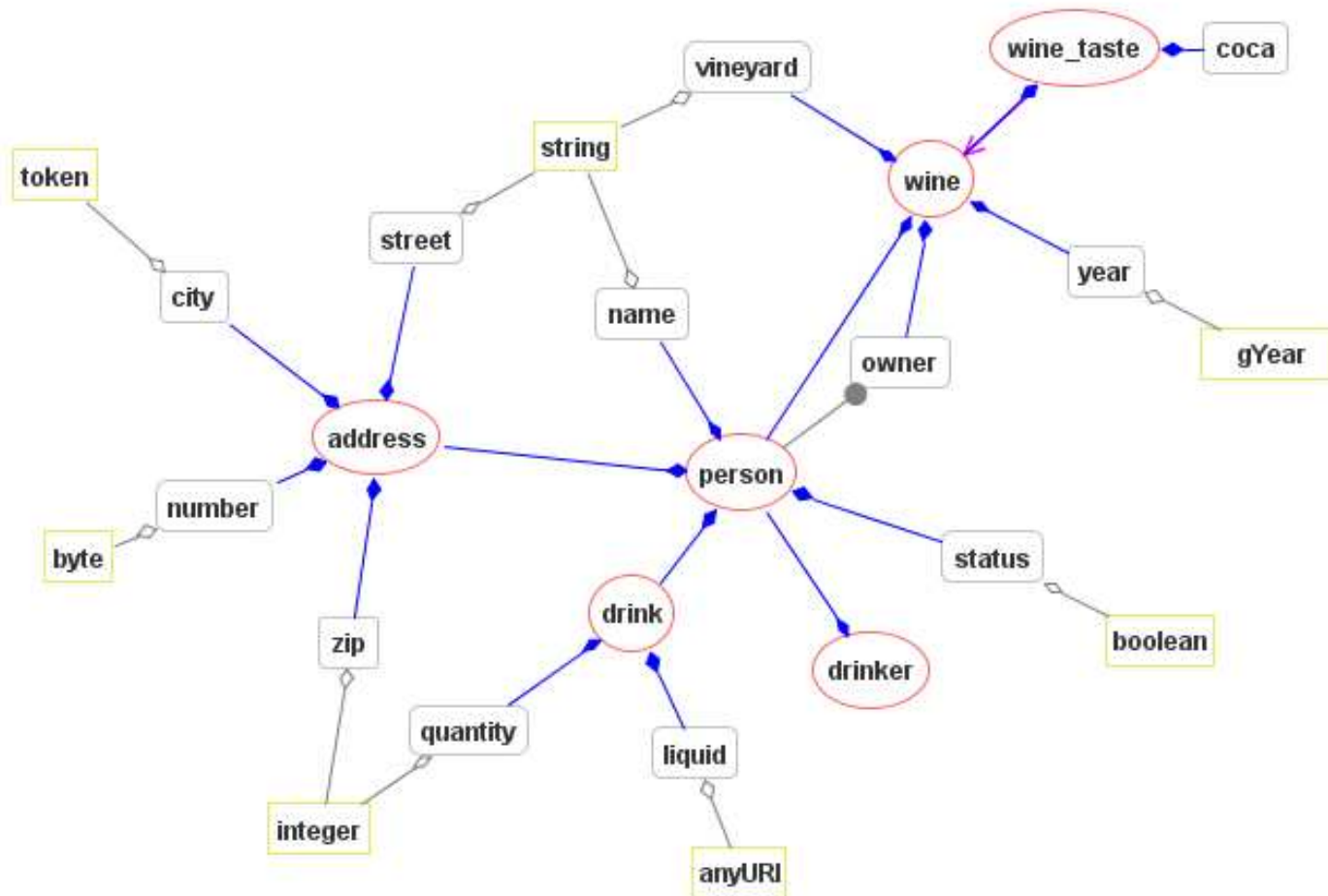


Generated by XmlSpy

www.altova.com

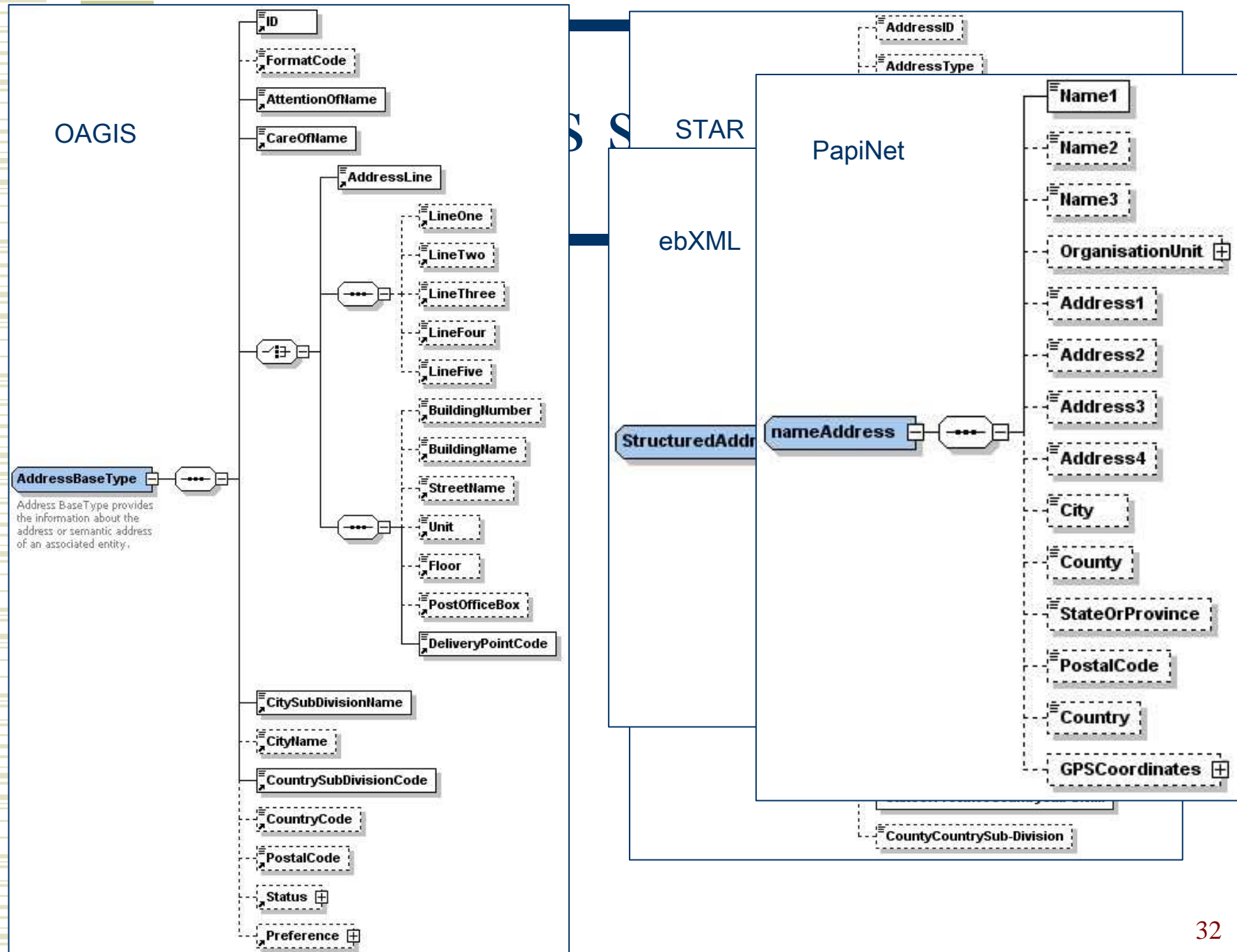
Classes	Object Properties of related classes	Object Datatypes of related Properties	Relationships
6 (wine_taste, wine, person, drinker, drink, address)	12 (quantity, vineyard, year, zip, status, city, coca, street, boolean, liquid, name, owner)	7 (string anyURI gYear token integer byte number)	Owner IS A Person Drinker IS A Person Coca DISJOINT Wine

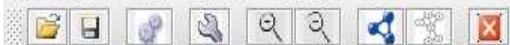
Janus : Ontologie dérivée



Cas d'utilisation B2B

- ◆ 75% des entreprises qui réalisent des échanges B2B utilisent des standards (source *E-Business W@tch Report, 2007*)
- ◆ Les organismes de normalisation B2B définissent leurs messages et leurs données par secteur d'activité
- ◆ Nous avons étudié 30 standards B2B (tourisme, commerce, assurance, finance, chimie,...)
 - Chacun d'entre eux propose des normes basées sur XML, sous forme XSD and DTD (nous avons recueilli déjà ~3000 fichiers)
 - Aucun ne fournit d'ontologie (!)





NMTOKEN

- URI
- additional_street_name
- address
- address_base
- address_identification
- address_information
- address_line
- agency_identification_code
- agency_identification_code_content
- alternative_communication_method
- altitude_measure
- amount
- anyURI
- attention
- attention_name
- block_name
- boolean
- build_name
- build_number
- care
- care_name
- city
- city_code
- city_name
- city_sub_division_name
- code
- comment
- communication_role
- contact_description
- contact_information
- contact_name
- contact_number
- container_description
- content
- coordinate
- coordinate_base

Graph Tag Cloud List Item Detail Property Lattice

Layout

- Hierarchical
- Fast Organic
- Organic
- Self-Organizing
- Compact Tree
- Radiatree
- Tree
- Reset

Graph

Init Graph

Show Whole

Actual Size

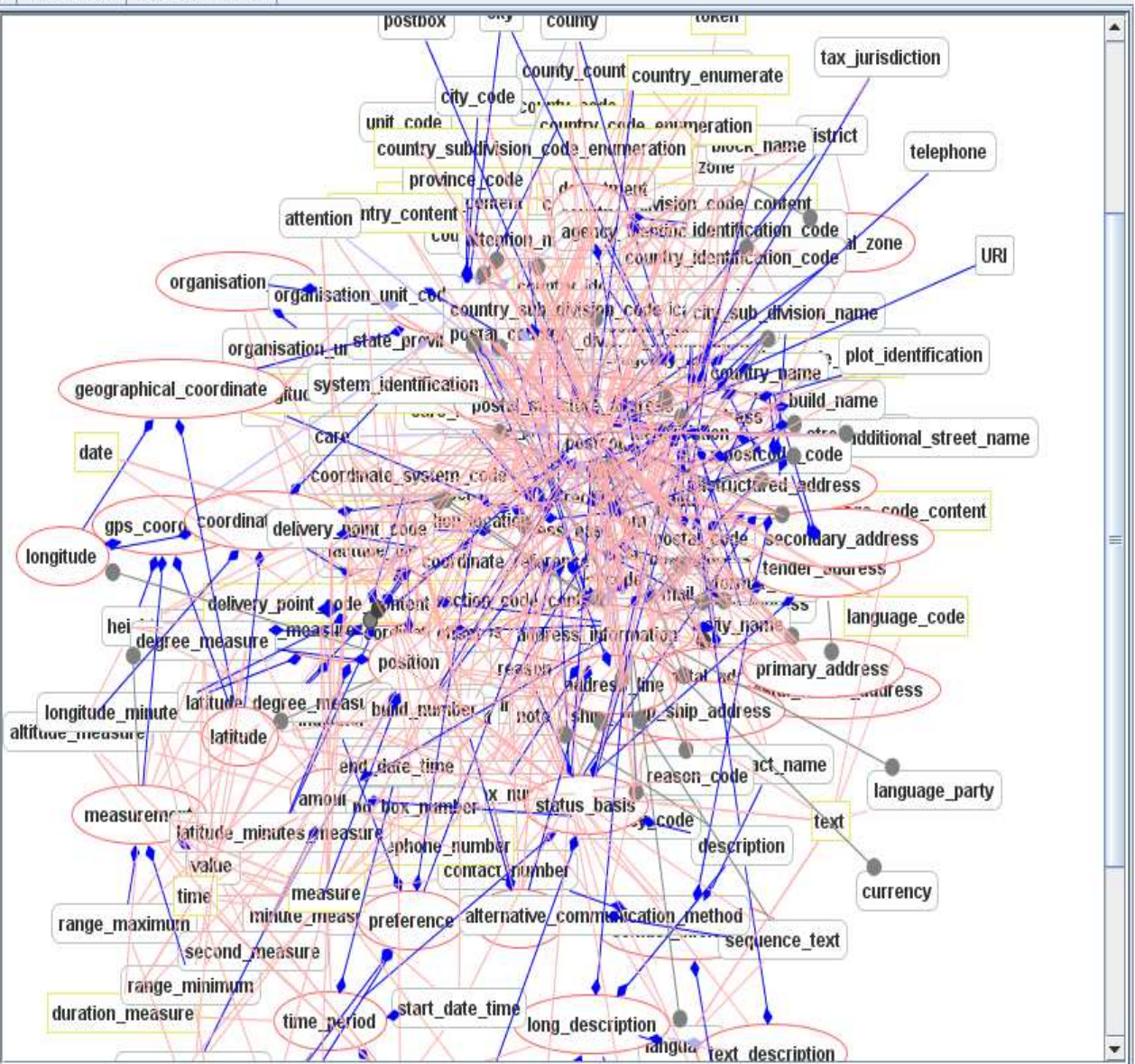
Fit Window

Relationships

- Show All
- Properties
- Shared T...
- Synonyms
- Related
- Data Type

Nodes

- Show All
- Classes



Loaded 195 concepts and 1296 edges

5. Conclusion

- ◆ Résumé
 - Puissance et intérêt des ontologies
 - Construction à partir de schémas : Janus
- ◆ Perspectives
 - Enrichir les règles de mapping xsd à Mod. Sém.
 - Améliorer l'interactivité
 - Améliorer les performances
 - Ajouter un langage déclaratif de « haut niveau »