

➤ Construction d'ontologie à partir de modèle conceptuel de données

CHOUMÉLÉ NANDONG Lucry Maël , ROUSSEY Catherine

UMR MISTEA, INRAE, Centre Occitanie-Montpellier, France

Atelier in-Ovive @ IC, pfiA 2025

1er juillet 2025

Introduction

Contexte

- Données Tabulées
 - Données Cachées
 - Données redondantes

Problématique

- Quelles sont les règles de transformation d'un modèle conceptuel de données en modèle graphe ?
- Générer un modèle graphe à partir d'un modèle conceptuel de données.
- Générer une ontologie à partir d'un modèle graphe.

Objectifs

- Définir les règles de traduction d'un modèle conceptuel de données en modèle graphe;
- Mettre sur pied une architecture de transformation utilisant l'outil Draw.IO d'un modèle conceptuel de données en UML en modèle graphe en CHOWLK.



Travaux antérieurs

Règles de traduction de BDR vers Ontologie:

1. Une table avec une clé primaire non composée est transformée en une classe OWL;
2. Une table avec une clé primaire composée de clés étrangères, dite table d'association, est transformée en deux propriétés objet dont l'une est l'inverse de l'autre;
3. Une clé primaire (PK) est transformée en une propriété fonctionnelle;
4. Une clé étrangère (FK) est transformée en une propriété d'objet;
5. Une relation hiérarchique est transformée en une relation de sous-classe (subClassOf);
6. Une colonne est transformée en une propriété de données;
7. Une contrainte Not NULL sur une colonne est transformée en une contrainte de cardinalité OWL;
8. Une contrainte UNIQUE sur une colonne transforme la propriété associée en une propriété fonctionnelle.



Travaux antérieurs

Listes des travaux:

Listes des travaux	Auteur	Année	Entrée	Sortie	Règles
R2O, an Extensible and Semantically Based Database to ontology Mapping Language	Jesús Barrasa et al.	2004	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	3(1, 4 ,6)
Database to Ontology Mapping Generation for Semantic Interoperability	Raji Ghawi et al.	2007	interroge une BDR	une ontologie non peuplée (fichier owl)	6
Direct Migration Method of RDB to Ontology while Keeping Semantics	Jamal Bakkas et al.	2013	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	6
Architecture for Mapping Relational Database to OWL ontology: An approach to enrich ontology terminology Validated with Mutation Test	Cristiane A.G. Huve et al.	2019	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	6
Mapping Relational Database to OWL Ontology Based on MDE Settings	Benamar Bouougada et al.	2021	une BD exportée avec des fichiers sql	une ontologie peuplée (fichier owl)	6
A novel approach for learning ontology from relational database: from the construction to the evaluation	Bilal Ben Mahria et al.	2021	une BD exportée avec des fichiers sql	une ontologie peuplée (fichier owl)	6



Travaux antérieurs

Listes des travaux:

Learning ontology from relational DataBase	Man Li et al.	2005	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	8
Automatic Generation of Ontology Based on Database	Lei ZHANG et al.	2011	interroge une BDR	une ontologie peuplée (fichier owl)	8
Mapping Relational Databases into OWL Ontology	Haiyun Ling et al.	2014	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	8
An Approach For Transforming of Relational Databases to OWL Ontology	Mona Dadjoo et al.	2015	une BD exportée avec des fichiers sql	une ontologie non peuplée (fichier owl)	8
automatic conversion of relational databases into ontologies: a comparative analysis of protégé plug-ins performance	Kgotatso Desmond Mogotlane et al.	2016	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	7
Integration Mapping Rules: Transforming Relational Database to Semantic Web Ontology	Mohamed A. G. Hazber et al.	2016	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	8
Ontology learning from relational databases	Batool Lakzaei et al.	2021	connexion à une BDR avec une API	une ontologie peuplée (fichier owl)	8



Travaux antérieurs

Remarques

- Ces travaux précédemment mentionnés opèrent directement à partir d'une base de données relationnelle ou d'un fichier SQL pour la traduire automatiquement sans intervention humaine;
- Ces approches ne permettent pas de réutiliser des ontologies de référence pour créer une nouvelle ontologie;
- Les articles utilisant les règles 7 et 8 reposent sur une logique du monde clos, héritée des bases de données.



Architecture

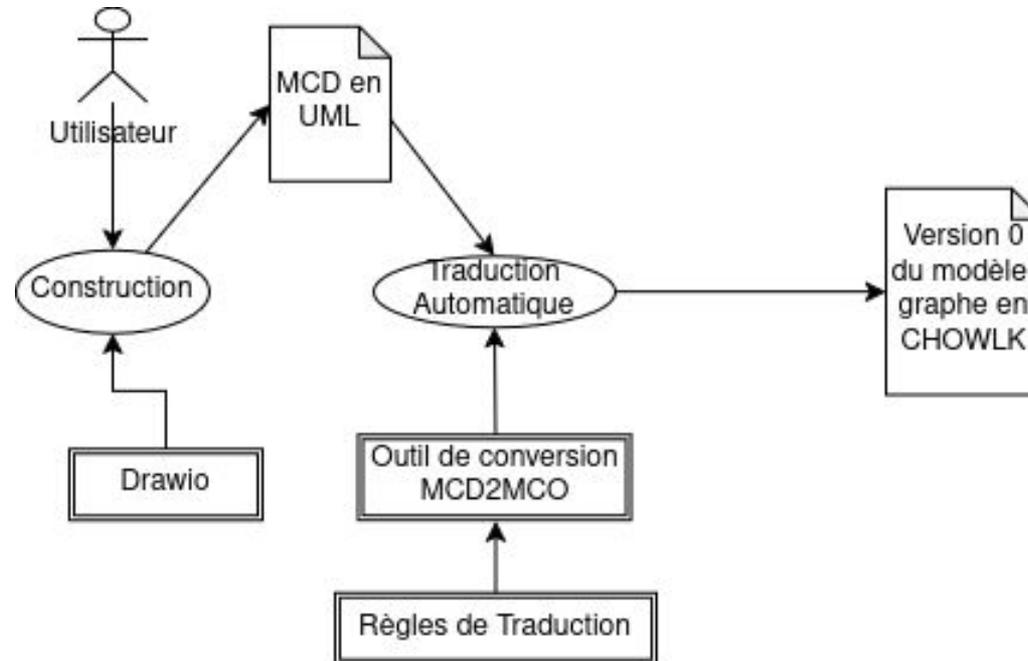


Figure : Schéma représentant le processus de transformation d'un MCD en modèle graphe.

Méthodologie

Règles de traduction des modèles

1. E2C: Entité vers Classe;

Le nom de l'entité est utilisé comme nom de la classe. Ainsi, son URI se compose du préfixe et du nom de l'entité (**prefix:NomEntité**).



Méthodologie

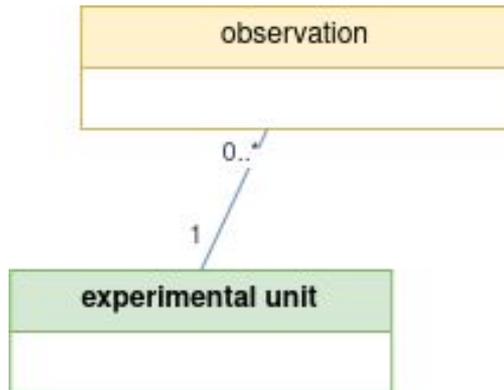
Règles de traduction des modèles

2. A2OP : Association vers propriété d'objet;

Création de deux propriété d'objet inverse l'une de l'autre

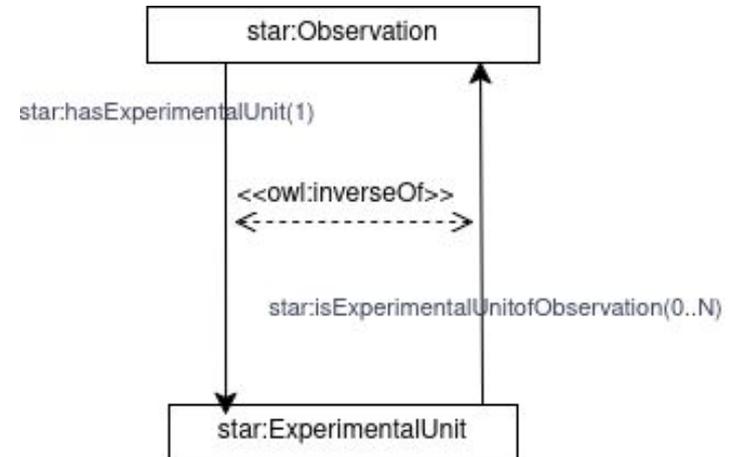
De la classe A vers la classe B

Création d'une propriété d'objet qui a pour convention de nommage **prefix:hasNomB**



De la classe B vers la classe A

Création d'une propriété d'objet qui a pour convention de nommage **prefix:isNomAOfNomB**

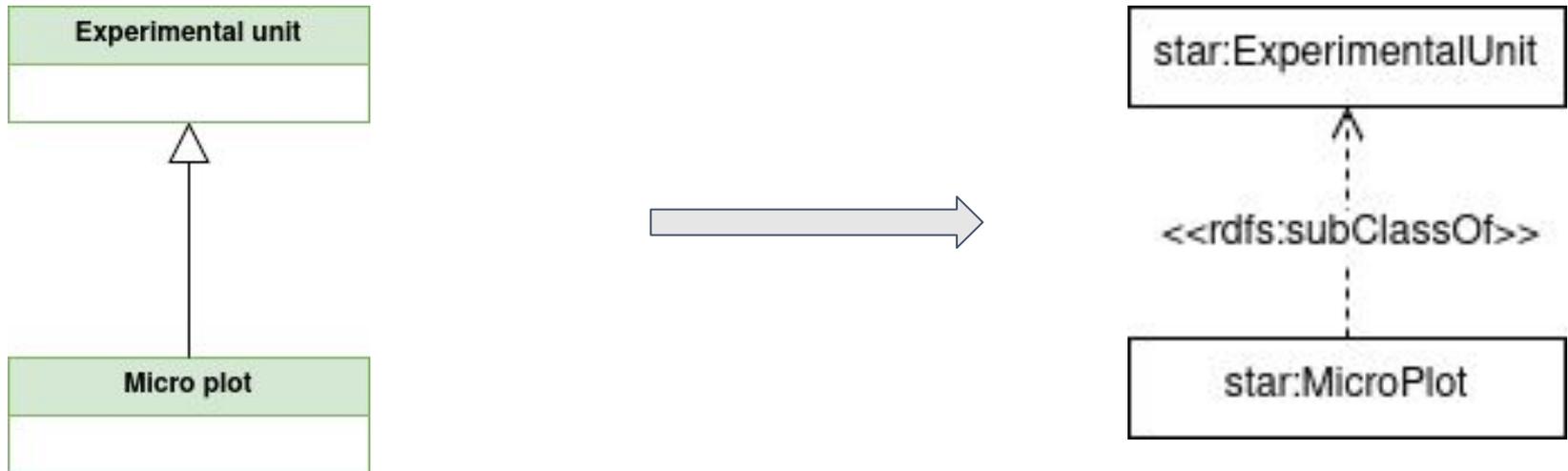


Méthodologie

Règles de traduction des modèles

3. I2S: Inclusion de clés primaires vers relation SubclassOf;

si T est une spécialisation de l'entité A alors T sera convertie en une sous-classe de la classe correspondant à l'entité A. Et son URI sera constitué d'un Préfixe et du nom de l'entité **Prefix:NomEntité** de façon automatique, et sera relié avec la classe parente par la relation **rdfs: subClassOf**.

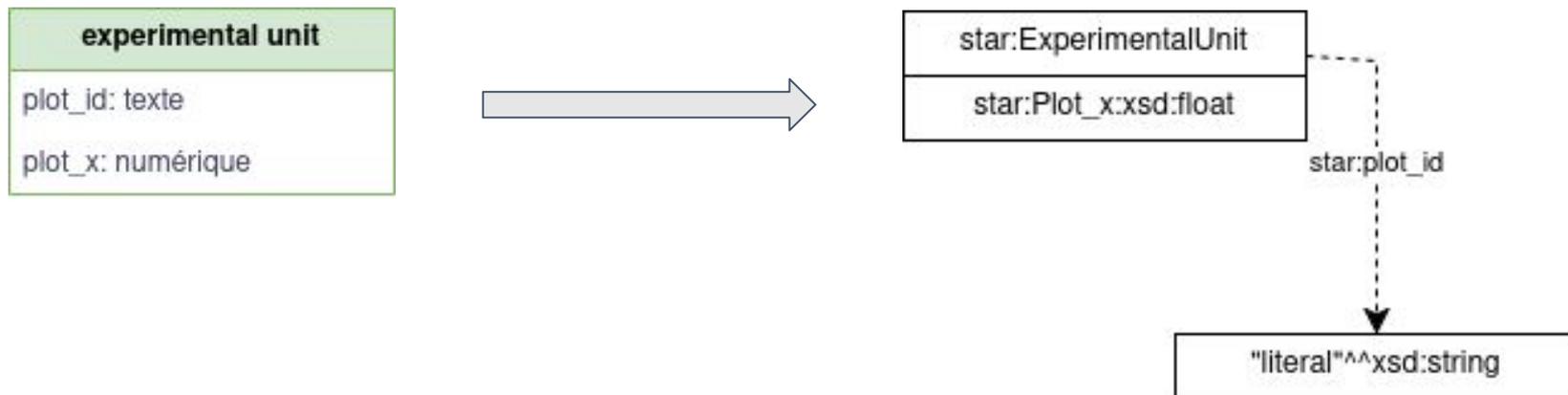


Méthodologie

Règles de traduction des modèles

4. A2DP: Attribut vers propriété de type de données;

Attribut devient une propriété de type de données ou propriété d'annotation.



Règles de traduction des modèles

A2DP: Attribut vers propriété de type de données;

Cas Particulier : clé Primaire, Not NULL, UNIQUE,...

- Si un attribut est clé Primaire elle devient une propriété fonctionnelle avec une cardinalité de 1 ($\text{minCardinality} = 1$ et $\text{maxCardinality} = 1$), Si cette clé primaire est de type number alors elle peut être abandonnée sinon elle peut être conservée;
- Si un attribut est Not NULL elle est transformé en une propriété OWL avec cardinalité minimale (minCardinality) = 1. Cela garantit que chaque instance de la classe possédant cette propriété doit obligatoirement avoir une valeur;
- Si un attribut est UNIQUE elle est transformé en propriété fonctionnelle (FunctionalProperty), elle est associée à une cardinalité maximale (maxCardinality) = 1;
- Si un attribut a pour type Numérique ou Nombre alors ce type se transforme en `xsd:float`;
- Si un attribut a pour type Texte ou Description alors ce type se transforme en `xsd:string`.



Outil MCD2MCO



<https://forge.inrae.fr/lucry-mael.choumel>

[e-nandong/mcd2mco](https://forge.inrae.fr/lucry-mael.choumel/e-nandong/mcd2mco)

Interface MCD2MCO

Entrez le Prefix de l'ontologie

Chargez un fichier

Drag and drop file here
Limit 200MB per file • XML

Browse files

Copie de STAR_modele_base2.drawio.xml 36.0KB

Lancer le traitement

Vérification terminée, les modifications seront appliquées.

Erreur de Relation entre personne et bbch_stage: texte

Modifications appliquées au fichier. Corriger les erreurs dans le fichier

Traitement terminé ! Voici le contenu du fichier généré :

Interface MCD2MCO

Entrez le Prefix de l'ontologie

Chargez un fichier

Drag and drop file here
Limit 200MB per file • XML

Browse files

STAR_modele_baseNew.drawio.xml 42.3KB

Lancer le traitement

Vérification terminée, les modifications seront effectuées.

Transformation en cours, les modifications seront effectuées.

Transformation en modèle Graphe Terminée. Modifications appliquées au Fichier

Traitement terminé ! Voici le contenu du fichier généré :

```
Contenu généré
<?xml version='1.0' encoding='UTF-8'?>
<mxfile host="app.diagrams.net" agent="Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:139.0)
Gecko/20100101 Firefox/139.0" version="27.1.5">
<diagram id="5G-F_DqTZ7vnh3qM1cWX" name="star_core_v0">
<mxGraphModel dx="3907" dy="1996" grid="1" gridSize="10" guides="1" tooltips="1" connect="1"
arrows="1" fold="1" page="1" pageScale="1" pageWidth="827" pageHeight="1169" math="0"
shadow="0">
<root>
<mxCell id="0" />
<mxCell id="1" parent="0" />
<mxCell id="2KE0voiH2TDJrb9lcf_v-1" value="star:Field"
style="rounded=0;whiteSpace=wrap;html=1;snapToPoint=1;points=[[0.1,0],[0.2,0],[1,0.9]];
parent="1" vertex="1">
```

Télécharger le fichier



INRAE

Construction d'ontologie à partir de modèle conceptuel de données
Atelier IN-OVIVEJuillet 2025 Dijon Lucry CHOUMELE

Evaluation

Production du MCD

- Les données STAR produit par Xavier DELPUECH de IFV;
- Les données ICASA produit par Farzaneh KAZEMI POUR de MISTEA.

Evaluation du modèle graphe

- Evaluation syntaxique avec le convertisseur CHOWLK Converter
- Test de l'outil MCD2MCO sur 2 cas d'usage
 - Le modèle du projet STAR ;
 - Le format d'échange ICASA.



Perspective

Architecture

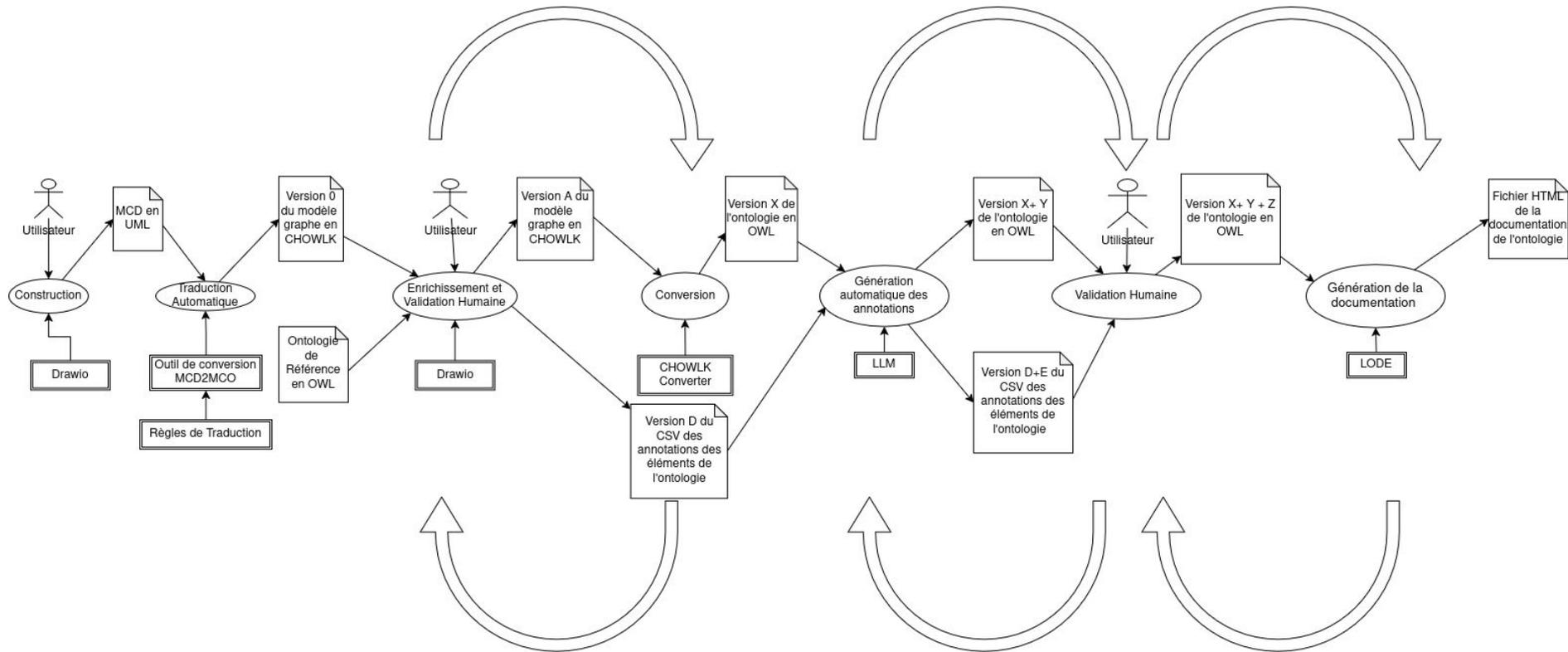


Figure : Schéma représentant le processus de transformation d'un MCD en une ontologie.

Conclusion

Nous avons proposé une méthode de transformation d'un MCD en modèle conceptuel graphe.

Nous avons développé un outil implémentant cette méthode.

Remerciements

Ce travail a été financé par une bourse #DigitAg.



Merci pour votre attention.



INRAE

Construction d'ontologie à partir de modèle conceptuel de données
Atelier IN-OVIVE Juillet 2025 Dijon Lucry CHOUMELE