

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

Lucry Maël CHOUMELE NANDONG<sup>1</sup>, Catherine ROUSSEY<sup>1</sup>

<sup>1</sup> UMR MISTEA, INRAE, Centre Occitanie-Montpellier, France

## Résumé

*Face aux défis de la sécurité alimentaire et du changement climatique, l'intégration sémantique des données agronomiques devient essentielle. Ces données, issues de sources variées, sont souvent stockées sous forme de bases relationnelles ou d'un ensemble de fichiers tabulés. Cet article présente une méthode de transformation de modèle conceptuel de données utilisant le langage graphique UML en modèle de graphe utilisant le langage CHOWLK. L'approche s'appuie sur l'outil CHOWLK converter [5] pour convertir le modèle de graphe en ontologie OWL. L'approche est semi-automatique combinant des étapes de traductions automatiques avec des étapes de validation et d'enrichissement manuelles.*

## Mots-clés

*modèle conceptuel de données, ontologie, UML, CHOWLK*

## Abstract

*Faced with the challenges of food security and climate change, the semantic integration of agronomic data is becoming essential. This data, from a variety of sources, is often stored in the form of relational databases or a set of tabulated files. This article presents a method for transforming a conceptual data model using the UML graphical language into a graph model using the CHOWLK language. The approach uses the CHOWLK converter tool [5] to convert the graph model into an OWL ontology. The approach is semi-automatic, combining automatic translation steps with manual validation and enrichment steps.*

## Keywords

*conceptual data model, ontology, UML, CHOWLK*

## 1 Introduction

Dans un contexte marqué par les enjeux croissants de sécurité alimentaire, d'adaptation au changement climatique et de transition vers une agriculture plus durable, la valorisation des données issues d'expérimentations agronomiques devient un levier stratégique. Ces données, issues de dispositifs expérimentaux variés sont souvent stockées sous forme d'ensemble de fichiers tabulés ou au mieux de bases de données relationnelles. La faible documentation associée à ces données rend difficile leur interprétation et réutilisation en dehors des personnes qui ont produit ces données. Afin de rendre ces données interopérables, les technologies du Web Sémantique proposent des langages de structuration des données sous forme de graphes (RDF, OWL,

SKOS) ainsi que des protocoles de publication et d'échange de données par le biais du Web. Les données ainsi disponibles sur le Web deviennent des graphes de connaissances, et leurs schémas des ontologies. Dans ce travail, nous nous intéressons à la transformation des modèles conceptuels de données utilisant le langage UML, en modèle de graphes utilisant le langage CHOWLK [5]. CHOWLK est une extension d'UML dédiée aux ontologies du Web sémantique. Nous proposons une méthode semi-automatique combinant l'utilisation d'outils tels que Drawio et CHOWLK Converter [4], des modèles de langues via Mistral [9] et l'expertise humaine. Cette méthode facilite la génération et la mise à jour continue d'ontologies à partir de modèles conceptuels existants.

## 2 Travaux antérieurs

Des travaux sur la construction d'ontologie à partir de bases de données relationnelles (BDR) ont proposé des règles de transformation. Ces règles définissent des correspondances entre les éléments des bases relationnelles et les éléments d'une ontologie peuplée ou non. Au total, nous avons trouvé 8 règles de transformation dans la littérature :

1. Une table avec une clé primaire non composé 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.

Plusieurs travaux de recherche ont mis en œuvre ces règles dans des systèmes de conversion différents.

## 2.1 Approches basées sur les règles 1 à 6 de transformation

Certaines méthodes de transformation automatique de bases de données relationnelles en ontologies OWL se limitent à l'application des règles 1 à 6, généralement centrées sur la génération de la TBox (structure conceptuelle de l'ontologie), sans prise en charge explicite des cardinalités ou des axiomes complexes.

En 2007, les travaux de [7] proposent un outil de conversion automatique de base de données relationnelle en ontologie intitulé DB2OWL, basé sur une approche en deux étapes. Premièrement, le contenu de chaque source d'information est décrit par une ontologie locale. Deuxièmement, les ontologies locales sont ensuite intégrées dans une ontologie globale. Ainsi, cette approche crée des correspondances entre les classes des différentes ontologies locales. Le prototype a été testé sur trois cas d'usage.

Sur une logique similaire les travaux de [18] proposent en 2011 un outil, OGSRD, de conversion de BDR en ontologie OWL de manière automatique. Leur méthode se déroule en trois étapes : extraction des tables, relations et instances de la base de données ; transformation des structures à l'aide des règles de traduction ; et génération de l'ontologie cible peuplée. Bien que l'outil produise une ontologie exploitable dans Protégé, un travail reste nécessaire pour améliorer la qualité finale de l'ontologie.

Dans la continuité de cette approche, [2] propose une méthode de transformation directe opérant à deux niveaux : au niveau du schéma, les tables sont converties en classes OWL, les attributs en propriétés de type de données, et les clés primaires ou étrangères en propriétés fonctionnelles ou objets ; au niveau des données, chaque enregistrement devient un individu OWL nommé, les liens entre entités étant préservés via les clés étrangères. Cette approche systématique assure une correspondance entre les modèles relationnels et ontologiques.

Enfin en 2021, Ben Mahria et al. [3] proposent une méthode qui extrait à la fois les tables et les instances d'un fichier SQL pour avoir une ontologie finale peuplée. La phase de développement génère l'ontologie en deux parties : la TBox (Tables et relations) et l'ABox (instances). Une évaluation a été menée en comparant les ontologies générées à une ontologie de référence (GoodRelations), avec des résultats positifs, notamment dans le contexte des bases e-commerce.

## 2.2 Approches basées sur les règles 1 à 8 de transformation

Plusieurs travaux ont proposé des méthodes automatiques de transformation de bases de données relationnelles en ontologies OWL, reposant les règles 1 à 8 de transformations, couvrant les classes, propriétés, hiérarchie, cardinalité et instances.

Parmi ces travaux, en 2005 [12] introduit une méthode directe de transformation sans passer par un modèle intermédiaire. Leur approche permet de générer l'ensemble complet d'une ontologie OWL (TBox et ABox) et a été validée sur le projet de bibliothèque numérique de l'Université

Renmin de Chine, couvrant 30 000 ressources et aboutissant à 20 classes, 400 propriétés et 30 000 instances. De manière complémentaire, en 2021 [10] étendent ce même cadre pour traiter des cas plus complexes tels que l'héritage multiple, les entités faibles ou les attributs partagés entre plusieurs tables. Leur méthode montre de bonnes performances sur différents jeux de données, prouvant sa robustesse.

En 2015, [6] propose également une méthode automatique pour transformer une base de données relationnelle en ontologie OWL, structurée en trois étapes : extraction des informations de la base (tables, colonnes, contraintes, triggers) ; création d'un modèle conceptuel intermédiaire à partir des informations précédentes ; puis génération de l'ontologie finale à l'aide de règles de traduction. Une particularité de cette approche est la prise en compte des triggers pour les traduire en contraintes. Cependant, l'ontologie générée intègre l'hypothèse du monde clos<sup>1</sup> ce qui ajoute des contraintes à l'ontologie. Nous souhaitons produire les ontologies intégrant l'hypothèse du monde ouvert<sup>2</sup>.

Par ailleurs, plusieurs outils ont été conçus pour automatiser ce processus. DataMaster [1] est un plug-in de Protégé qui importe automatiquement la structure et les données d'une base relationnelle via JDBC/ODBC pour générer une ontologie OWL. OntoBase [11], quant à lui, transforme le schéma d'une base relationnelle en ontologie OWL. Il permet la création de classes à partir des colonnes et l'interrogation directe de la base. En 2016, [13] compare ces deux outils sur une base Oracle en évaluant leur conformité à l'ensemble des règles mentionnées. Ils concluent qu'OntoBase est plus précis, respectant mieux les relations entre entités et générant moins d'erreurs que DataMaster qui produit plus de propriétés erronées.

Enfin, en 2016 [8] décrit une méthode similaire implémentée avec Apache Jena en Java et MySQL, illustrée par un exemple pratique et validée à l'aide de validateurs d'ontologie. Mais comme l'approche de [6], l'ontologie produite intègre l'hypothèse du monde clos ce qui va à l'encontre du principe des ontologies du Web sémantique.

Des états de l'art comparent les approches étudiées. En 2011, [16] compare sept méthodes de transformation SQL vers OWL/RDFS, en soulignant l'importance des combinaisons entre clés primaires et étrangères, et les limites des transformations directes pour refléter la sémantique métier. Puis en 2023, [14] analyse treize approches selon la richesse des règles appliquées. Certaines, comme [12, 8], couvrent jusqu'à huit règles, tandis que d'autres, comme [7], se limitent à la transformation de la structure. Les méthodes récentes, dont celles de [3, 10], apparaissent comme plus complètes. Cet état de l'art conclut à l'absence de standard, et recommande de consolider les meilleures règles existantes.

1. (qui est l'hypothèse que ce qui n'est pas connu pour être vrai doit être faux. Ainsi ce qui est inscrit dans la base est vrai et ce qui n'est pas inscrit dans la base est faux)

2. ("ce n'est pas parce qu'on ne connaît pas une information que cette information est fausse")

La majorité des travaux précédemment mentionnés opèrent directement à partir d'une base de données relationnelle pour la traduire automatiquement sans intervention humaine. De plus, ces approches ne permettent pas de réutiliser des ontologies de référence pour créer une nouvelle ontologie spécialisant ces éléments. Les articles utilisant les règles 7 et 8 reposent sur une logique du monde clos, héritée des bases de données. Ces contraintes sont strictes, elles exigent que certaines conditions soient toujours respectées. Or, les ontologies OWL s'inscrivent dans une logique du monde ouvert favorisant ainsi l'intégration de données. La méthodologie Linked Open Terms (LOT) propose un cadre pour le développement d'ontologies [15]. Cette méthode se fait en quatre étapes : en premier la spécification des besoins à partir de cas d'usage et de questions de compétence, en deuxième l'implémentation à travers la modélisation conceptuelle, l'encodage en OWL et la réutilisation d'ontologies existantes, en troisième la publication de l'ontologie avec une documentation accessible et des URI persistants ; et en quatrième, la maintenance continue grâce à des outils de gestion collaborative. Cette méthode a été évaluée dans dix-huit projets réels, et s'appuie sur des outils reconnus tels que Protégé, Widoco, GitHub et OOPS!. Elle favorise la création d'ontologies réutilisables, publiables selon les principes FAIR. Nous partons de l'hypothèse que nous disposons d'un Modèle Conceptuel de Données (MCD). Cette hypothèse nous permet de proposer nos règles de transformation.

### 3 Règles de traduction

Nous allons travailler avec des modèles conceptuels de données représentant le contenu des bases de données relationnelles, ainsi nous proposons une reformulation des règles de transformation de l'état de l'art.

**E2C : Entité vers Classe** Cette règle indique qu'une entité du modèle conceptuel de la BDR est convertie en classe OWL. Le nom de l'entité est utilisé comme nom de la classe.

**I2S : Inclusion de clés primaires vers Relation de sous-classe** Cette règle indique que si l'entité A a comme clé primaire un sous-ensemble de la clé primaire de l'entité B, ou que A est une spécialisation de l'entité B, alors A sera convertie en une sous-classe de la classe correspondant à l'entité B.

**A2OP : Association vers Propriété d'Objet** Cette règle indique qu'une association entre 2 entités A et B dans le modèle conceptuel de la BDR est convertie en deux propriétés d'objet : Une pour relier les individus de la classe A vers les individus de la classe B et sa propriété inverse qui relie les individus de la classe B vers les individus de la classe A. L'association porte des cardinalités qui sont traduites en cardinalité OWL. L'utilisateur devra vérifier le nom de la propriété d'objet proposé automatiquement par l'outil.

**A2DP : Attribut vers Propriété de Données** Cette règle indique qu'un attribut devient une propriété de type de données ou propriété d'annotation. Le choix est laissé à l'utilisateur en fonction de l'usage du contenu de la propriété.

- Si un attribut est clé primaire, elle devient une propriété de données fonctionnelle avec une cardinalité de 1 (  $\text{minCardinality} = 1$  et  $\text{maxCardinality} = 1$  ). Si cette clé primaire est un nombre (clé primaire incrémentale) alors elle peut être abandonnée sinon elle peut être conservée.
- Si un Attribut est UNIQUE elle est transformée en propriété fonctionnelle (FunctionalProperty) Elle est associée à une cardinalité maximale ( $\text{maxCardinality} = 1$ ), ce qui signifie qu'un individu (instance d'une classe) ne peut avoir qu'une seule valeur pour cette propriété.

## 4 Description de l'approche

La figure 1<sup>3</sup> présente notre approche, qui débute par la construction d'un Modèle Conceptuel de Données (MCD) par l'utilisateur, à l'aide de l'outil Drawio et du langage UML. Ce modèle représente les données actuellement stockées sous forme tabulée. Le MCD est ensuite automatiquement traduit grâce à un outil de conversion et à un ensemble de règles que nous avons définies, afin de générer une version 0 du modèle graphe écrit à l'aide du langage graphique CHOWLK.

Le diagramme CHOWLK généré précédemment est ensuite enrichi grâce à des ontologies de référence et validé manuellement par l'utilisateur. À l'issue de cette phase, deux artefacts sont produits :

- un fichier CSV (version D) contenant les labels des éléments de l'ontologie,
- un modèle graphe (version A) écrit à l'aide du langage graphique CHOWLK.

Le modèle graphe (version A) est converti en un fichier OWL (version X) de l'ontologie grâce à l'outil CHOWLK Converter. Ce processus d'enrichissement, de validation et de conversion est répété en boucle, jusqu'à obtention d'une version du modèle graphe en CHOWLK (version X) considéré comme complet et du fichier CSV (version D) associé.

Le fichier CSV (version D) des annotations des éléments de l'ontologie et le modèle graphe en CHOWLK (version X) sont utilisés comme entrée par un grand modèle de langages (LLM). Le LLM génère automatiquement des propositions de définition pour chaque élément de l'ontologie et produit donc une nouvelle version du fichier CSV (version D + E) et de l'ontologie OWL (version X+Y). L'ontologie OWL est ensuite soumise à une validation humaine pour générer une version finale. Ce processus est également répété jusqu'à obtention d'une ontologie en OWL complète (version X+Y+Z).

Enfin, l'ontologie OWL (version X+Y+Z) est utilisée pour générer la documentation HTML de l'ontologie à l'aide de l'outil LODÉ. Ce processus est également itératif jusqu'à

3. Une version plus grande de l'image est disponible sur [https://forgemia.inra.fr/star\\_casdar/stagedigitag2025/-/blob/main/many\\_to\\_many-Page-10.png](https://forgemia.inra.fr/star_casdar/stagedigitag2025/-/blob/main/many_to_many-Page-10.png)

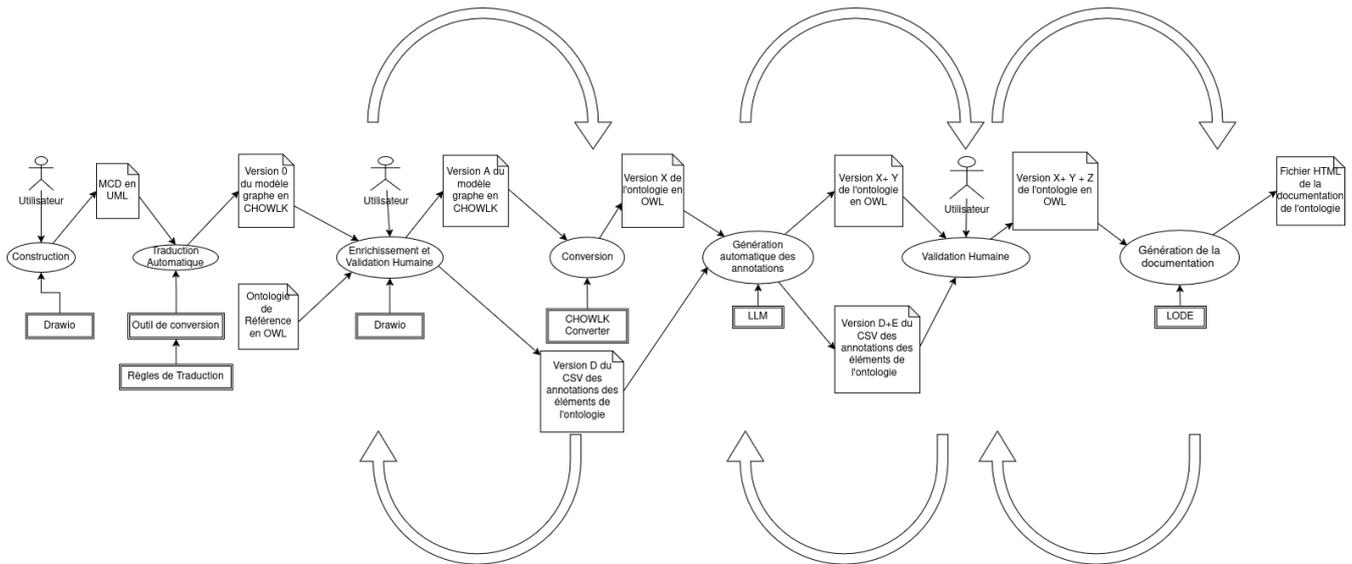


FIGURE 1 – Schéma représentant le processus de transformation d'un MCD vers une ontologie.

obtention d'une documentation HTML et d'une ontologie complète (version  $X+Y+Z$ ).

## 5 Conclusion

Nous avons présenté une méthode de transformation progressive d'un Modèle Conceptuel de Données (MCD) vers un modèle conceptuel de graphe pour produire une ontologie OWL, en combinant automatisations et validation humaine. Cette approche sera testée sur le format standard d'échange de données ICASA [17], utilisé en expérimentation agronomique. Notre travail portera principalement sur la première partie de l'architecture : la génération automatique du diagramme CHOWLK et son enrichissement via l'intervention humaine.

## 6 Remerciements

Ce travail a été financé par une bourse #DigitAg.

## Références

- [1] Csongor Nyulas. Datamaster. <https://protegewiki.stanford.edu/wiki/OntoBase>, 2008. Consulté en mai 2025.
- [2] Jamal Bakkas, Mohamed Bahaj, and Abderrahim Marzouk. Direct migration method of rdb to ontology while keeping semantics. *International Journal of Computer Applications*, 65(3), 2013.
- [3] Bilal Ben Mahria, Ilham Chaker, and Azeddine Zahi. A novel approach for learning ontology from relational database : from the construction to the evaluation. *Journal of Big Data*, 8(1) :25, 2021.
- [4] Serge Chávez-Feria, Raúl García-Castro, and María Poveda-Villalón. Converting uml-based ontology conceptualizations to owl with chowlk. In *European Semantic Web Conference*, pages 44–48. Springer, 2021.
- [5] Serge Chávez-Feria, Raúl García-Castro, and María Poveda-Villalón. Chowlk : from uml-based ontology conceptualizations to owl. In *European Semantic Web Conference*, pages 338–352. Springer, 2022.
- [6] Mona Dadjoo and Esmail Kheirkhah. An approach for transforming of relational databases to owl ontology. *arXiv preprint arXiv :1502.05844*, 2015.
- [7] Raji Ghawi and Nadine Cullot. Database-to-ontology mapping generation for semantic interoperability. In *Third international workshop on database interoperability (InterDB 2007)*, volume 91. Citeseer, 2007.
- [8] Mohamed AG Hazber, Ruixuan Li, Xiwu Gu, and Guandong Xu. Integration mapping rules : Transforming relational database to semantic web ontology. *Appl. Math*, 10(3) :1–21, 2016.
- [9] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv :2401.04088*, 2024.
- [10] Batool Lakzaei and Mehrnoush Shamsfard. Ontology learning from relational databases. *Information Sciences*, 577 :280–297, 2021.
- [11] Len Yabloko. Ontobase. <https://protegewiki.stanford.edu/wiki/OntoBase>, 2009. Consulté en mai 2025.
- [12] Man Li, Xiao-Yong Du, and Shan Wang. Learning ontology from relational database. In *2005 International conference on machine learning and cybernetics*, volume 6, pages 3410–3415. IEEE, 2005.
- [13] Kgotatso Desmond Mogotlane and Jean Vincent Fonou-Dombeu. Automatic conversion of relational databases into ontologies : a comparative analysis of prot\`eg\`e plug-ins performances. *arXiv preprint arXiv :1611.02816*, 2016.

- [14] Rosalba Mosca, Massimo De Santo, and Rosario Gaeta. Ontology learning from relational database : a review. *Journal of ambient intelligence and humanized computing*, 14(12) :16841–16851, 2023.
- [15] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. Lot : An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111 :104755, 2022.
- [16] Juan F Sequeda, Syed Hamid Tirmizi, Oscar Corcho, and Daniel P Miranker. Survey of directly mapping sql databases to the semantic web. *The Knowledge Engineering Review*, 26(4) :445–486, 2011.
- [17] Jeffrey W White, LA Hunt, Kenneth J Boote, James W Jones, Jawoo Koo, Soonho Kim, Cheryl H Porter, Paul W Wilkens, and Gerrit Hoogenboom. Integrated description of agricultural field experiments and production : The icasa version 2.0 data standards. *Computers and electronics in agriculture*, 96 :1–12, 2013.
- [18] Lei Zhang and Jing Li. Automatic generation of ontology based on database. *Journal of Computational Information Systems*, 7(4) :1148–1154, 2011.