POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Développement d'un cadre d'interopérabilité pour l'intégration verticale ERP/MES : OntoIMR pour les gammes de fabrication

CLAIRE JODEAU
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie industriel

Août 2025

© Claire Jodeau, 2025.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Développement d'un cadre d'interopérabilité pour l'intégration verticale ERP/MES : OntoIMR pour les gammes de fabrication

présenté par **Claire JODEAU**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Robert PELLERIN, président
Christophe DANJOU, membre et directeur de recherche
Samir LAMOURI, membre externe

REMERCIEMENTS

Je souhaite tout d'abord exprimer ma profonde gratitude à mon directeur de recherche, Christophe Danjou, pour son accompagnement tout au long de ces deux années de maîtrise. Ses conseils avisés, sa disponibilité et son soutien ont été déterminants dans la réalisation de ce mémoire.

Je tiens également à remercier chaleureusement les membres du jury, Robert Pellerin et Samir Lamouri, pour le temps qu'ils ont consacré à l'évaluation de ce travail et pour la pertinence de leurs retours écrits et oraux, d'autant plus appréciés en cette période estivale.

Mes remerciements vont également à toutes les personnes qui m'ont permis d'obtenir des retours précieux sur les différents logiciels étudiés. Je pense en particulier à Florent Tonson, dirigeant d'Isitecc, pour le temps qu'il m'a accordé et pour nos échanges toujours éclairants.

Enfin, je souhaite exprimer toute ma reconnaissance à mon entourage et à mes colocataires, qui ont partagé cette expérience avec moi et m'ont soutenu tout au long de cette aventure.

RÉSUMÉ

Dans un contexte où les exigences de flexibilité, de traçabilité et de réactivité deviennent centrales pour les entreprises manufacturières, l'intégration cohérente des données entre les systèmes de gestion de production (MES) et les progiciels de gestion intégrés (ERP) représente un enjeu stratégique majeur. L'essor de l'industrie 4.0, caractérisé par l'interconnexion croissante des systèmes informatiques et l'automatisation des processus, impose une circulation fluide de l'information entre les différents niveaux de gestion, du plancher de production jusqu'aux fonctions de planification stratégique. Pourtant, malgré les avancées technologiques et la normalisation progressive de certains échanges, de nombreuses entreprises continuent de rencontrer des difficultés liées à l'hétérogénéité des structures de données, à la redondance des saisies et au manque d'alignement sémantique entre leurs logiciels ERP et MES. Ces lacunes nuisent à la cohérence globale des flux d'information, freinent la transformation numérique et limitent le potentiel de pilotage intégré des opérations industrielles. C'est dans ce cadre que s'inscrit ce mémoire, qui propose une solution basée sur une ontologie formalisée des gammes de fabrication, jouant le rôle de modèle pivot entre les deux systèmes.

Le travail de recherche s'est organisé autour de deux axes. Le premier consistait à définir un modèle de données commun aux deux types de logiciels, en analysant les structures utilisées dans plusieurs ERP (SAP, Business Central, Divalto, JobBoss) et MES (SmartPortal, Sepasoft), ainsi que les recommandations de la norme ISA-95. Cette analyse comparative a permis de dégager un ensemble cohérent de concepts, structuré autour d'éléments essentiels tels que les opérations, les ressources, les séquences, les inspections et les composants matériels. Le second axe visait à formaliser ce modèle sous forme d'une ontologie exprimée en OWL DL, un langage standardisé du Web sémantique permettant une représentation explicite, logique et exploitable des connaissances.

La méthodologie adoptée comprend la modélisation manuelle dans Protégé, un processus de validation structurelle à l'aide d'outils de raisonnement sémantique, puis une validation sémantique à partir de deux cas d'étude industriels concrets. Ces cas portaient sur la production d'un meuble dans un ERP et d'un vélo dans un MES. L'ontologie a montré une capacité satisfaisante à représenter les concepts réels de chaque système, avec un taux de couverture supérieur à 85%. Afin de démontrer le caractère opérationnel de la solution, deux preuves de concept ont été mises en œuvre. La première a consisté à alimenter automatiquement l'ontologie à partir d'une base SQL issue d'un MES. La seconde a permis, à partir d'un peuplement manuel de l'ontologie avec des données ERP, de générer une structure de base de données

MES exploitable. Ces expérimentations ont été réalisées à l'aide de scripts Python dédiés.

Les résultats obtenus confirment la pertinence du modèle ontologique proposé, baptisé OntoIMR. Celui-ci permet de centraliser et de structurer les informations de gamme de manière cohérente, tout en assurant la possibilité d'une utilisation ascendante et descendante des données entre systèmes. L'ontologie facilite ainsi la réconciliation des données hétérogènes, réduit la duplication des saisies, et améliore la traçabilité des opérations de fabrication. Elle constitue une base prometteuse pour le développement futur de plateformes d'interopérabilité capables d'assurer un couplage standardisé entre ERP et MES, en lien avec des technologies d'orchestration comme les API, les ESB ou les modèles B2MML et OPC UA. Ce mémoire propose ainsi une avancée concrète vers une gouvernance intégrée de l'information manufacturière dans un contexte d'industrie 4.0.

ABSTRACT

In the context of digital transformation and Industry 4.0, manufacturing enterprises face growing pressure to increase responsiveness, ensure product traceability, and integrate operations across all levels of their value chain. Central to this challenge is the ability to enable seamless communication between Enterprise Resource Planning (ERP) systems and Manufacturing Execution Systems (MES), which are responsible for strategic and operational management, respectively. Despite technological advancements and the introduction of interoperability standards, many organizations still struggle to synchronize these systems effectively. This misalignment is particularly acute at the level of manufacturing routings, where inconsistent data structures, semantic gaps, and duplicated entries continue to hinder coordination and data reliability. This thesis addresses this issue by developing and validating a formal ontology that serves as a shared, structured representation of manufacturing routings between ERP and MES systems.

The research approach followed two main directions. First, a comparative study was conducted on the internal data structures of four ERP systems and two MES platforms, as well as the ISA-95 reference model. From this analysis, a unified set of manufacturing concepts was derived, including operations, sequences, production resources, inspections, and material inputs. These were used to construct an ontological model in OWL DL using the Protégé development environment, enabling a logically consistent, semantically rich, and software-agnostic representation of routing data.

Validation of the model was carried out through both structural and semantic testing. Structural validation ensured internal consistency using reasoning tools, while semantic validation tested the model's expressiveness on two real-world case studies: one describing the production of a bookcase in an ERP system, and the other the manufacture of a bicycle in a MES environment. The ontology demonstrated a high degree of coverage for both cases, confirming its ability to faithfully represent diverse system realities.

To assess the model's practical applicability, two proof-of-concept implementations were developed. In the first, the ontology was automatically populated from a SQL database exported from the MES. In the second, the ontology was manually populated with ERP data and then used to regenerate a compatible MES database structure. These demonstrations confirmed the model's capacity to mediate bidirectional data exchanges between systems and to serve as a pivot for future integrations.

The OntoIMR ontology developed in this research provides a robust foundation for ERP–MES

interoperability. It reconciles heterogeneous data into a coherent semantic model, reduces redundant inputs, and enhances the traceability of manufacturing processes. Beyond academic validation, it lays the groundwork for industrial applications such as middleware platforms, semantic data bridges, and orchestration services using standards like OPC UA and B2MML. As such, this thesis contributes a concrete step toward integrated information governance within smart manufacturing environments.

TABLE DES MATIÈRES

REMERCIEMENTS iii

RÉSUMÉ iv

ABSTRACT vi

LISTE DES TABLEAUX x

LISTE DES FIGURES xi

LISTE DES SIGLES ET ABRÉVIATIONS xii

LISTE DES ANNEXES xiii

CHAPITRE 1 INTRODUCTION 1

CHAPITRE 2 REVUE DE LITTÉRATURE 3
- 2.1 Industrie 4.0 3
- 2.2 MES, ERP, leur rôle dans l'industrie 4.0 5
  - 2.2.1 MES et ERP 5
  - 2.2.2 Les approches pour l'intégration verticale entre MES et ERP 6
- 2.3 Choix d'architecture de modèle de gamme de fabrication 9
  - 2.3.1 Sélection des critères d'évaluation 9
  - 2.3.2 Évaluation des modèles de données 13
- 2.4 Les modèles ontologiques dans le manufacturier 15
- 2.5 Revue critique 17
- 2.6 Conclusion 18

CHAPITRE 3 MÉTHODOLOGIE 19
- 3.1 Objectif de recherche 19
- 3.2 Méthodologie de recherche 20
  - 3.2.1 Développement du modèle commun de gamme 21
  - 3.2.2 Développement de l'ontologie 23
  - 3.2.3 Mise en œuvre 25

CHAPITRE 4 CONSTRUCTION DU MODÈLE COMMUN DE GAMME DE FABRICATION 26

4.1 Développement du modèle commun 26
4.1.1 Modèle commun MES 28
4.1.2 Modèle commun ERP 30
4.1.3 Modèle Commun 33
4.2 Construction de l'ontologie 36
4.2.1 Choix des caractéristiques de l'ontologie 36
4.2.2 Développement ontologique 39
4.3 Conclusion 44

CHAPITRE 5 VALIDATION 48
5.1 Méthodologie 48
5.2 Validation structurelle 48
5.2.1 Outils utilisés pour la validation structurelle 49
5.2.2 Résultats de la validation structurelle 49
5.2.3 *Domain* ou *Range* manquant dans les propriétés 50
5.2.4 Relation inverse non déclarée 51
5.2.5 Annotation manquante 51
5.2.6 Conclusion 52
5.3 Validation sémantique 52
5.3.1 Données utilisées pour la validation sémantique 52
5.3.2 Résultats de la validation sémantique 53
5.3.3 Conclusion 56
5.4 Preuves de concept 56
5.4.1 PoC1 — Intégration depuis les logiciels ERP et MES vers l'ontologie 56
5.4.2 PoC2 — Intégration depuis l'ontologie vers la base de données du MES SmartPortal 61
5.5 Conclusion 66

CHAPITRE 6 CONCLUSION 67
6.1 Synthèse des travaux / Summary of Works 67
6.2 Limitations de la solution proposée / Limitations 68
6.3 Améliorations futures / Future Research 69
6.4 Conclusion générale 69

ANNEXES 71

LISTE DES TABLEAUX

Tableau 2.1 Comparaison des systèmes de gestion de données 14
Tableau 4.1 Tableau comparatif des langages ontologiques 39
Tableau 4.2 Decription d'une partie des *Classes* et de leur *Data properties* de l'ontologie 40
Tableau 4.3 Descriptif des Object Properties liés à *Resource* 43
Tableau 4.4 Descriptif de l'ontologie 45
Tableau 5.1 Synthèse de l'analyse du scanner OOPS ! 50
Tableau 5.2 Taux de couverture de l'ontologie 53
Tableau A.1 Comparaison des systèmes de gestion de données justification 82
Tableau B.1 Descriptif des classes de l'ontologie Protege 90
Tableau C.1 Descriptif des Object Properties 92

LISTE DES FIGURES

Figure 2.1 Schéma de la hiérarchie fonctionnelle, (*Enterprise-control system integration. Part 1*, 2000) 6
Figure 3.1 Schéma de la méthodologie globale 20
Figure 4.1 Schéma de communication verticale idéale 26
Figure 4.2 Schéma de la méthodologie du développement du modèle commun de gamme 26
Figure 4.3 Schéma de la méthode du développement des modèles communs MES et ERP 27
Figure 4.4 Nombre de champs de données en fonction des sources MES 31
Figure 4.5 Nombre de champs de données en fonction des sources ERP 33
Figure 4.6 Nombre de champs de données du modèle commun 35
Figure 4.7 Schéma de la méthodologie du développement de l'ontologie 36
Figure 4.8 Interface du logiciel *Protégé* sur la vue des *Classes* pour les ressources 41
Figure 4.9 Interface du logiciel *Protégé* sur la vue des *DataProperty* des *Resources* 41
Figure 4.10 Interface du logiciel *Protégé* sur la vue des *Object Properties* des *Resource* 43
Figure 4.11 Visualisation de l'Ontograf centré sur les *Resource* 44
Figure 4.12 Interfaces de Protégé sur les Classes 45
Figure 4.13 Interfaces de Protégé sur les *Data Properties* 46
Figure 4.14 Interfaces de Protégé sur les *Object Properties* 46
Figure 4.15 Vue globale (OntoGraf) de l'ontologie, représentation simplifiée avec concepts et liens 47
Figure 5.1 Schéma de communication verticale partielle, des logiciels vers OntoIMR 57
Figure 5.2 Schéma des sources des données pour la communication verticale partielle 58
Figure 5.3 Preuves de concept 1 pour e-prélude 59
Figure 5.4 Preuves de concept 1 pour SmartPortal 60
Figure 5.5 Schéma de communication verticale partielle, d'OntoIMR vers les logiciels 62
Figure 5.6 Schéma des sources des données pour la communication verticale partielle 62
Figure 5.7 Preuve de concept 2 64

LISTE DES SIGLES ET ABRÉVIATIONS

ERP	Enterprise Resource Planning
MES	Manufacturing Execution System
ISA-95	International Society of Automation - Norme 95
OWL	Web Ontology Language
OWL DL	Web Ontology Language Description Logic
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
EAI	Enterprise Application Integration
ETL	Extract Transform Load
SOA	Service-Oriented Architecture
ESB	Enterprise Service Bus
API	Application Programming Interface
UML	Unified Modeling Language
PRT	Production Resource Tool
BOM	Bill of Materials
CSV	Comma-Separated Values
SQL	Structured Query Language
IPaaS	Integration Platform as a Service
CPS	Cyber-Physical System
IoT	Internet of Things
JSON	JavaScript Object Notation
HTTP	HyperText Transfer Protocol
SOAP	Simple Object Access Protocol
REST	Representational State Transfer
XSD	XML Schema Definition
DL Query	Description Logic Query
OOPS !	OntOlogy Pitfall Scanner !

LISTE DES ANNEXES

Annexe A Tableau comparatifs des bases de données 71
Annexe B Tableau descriptif des *Classes* de l'ontologie 83
Annexe C Tableau descriptif des *Object Properties* de l'ontologie 91

CHAPITRE 1 INTRODUCTION

L'essor de l'Industrie 4.0 transforme en profondeur les modes de production et de gestion industrielle. L'intégration et la circulation fluide de l'information entre les différents systèmes informatiques deviennent des enjeux cruciaux pour les entreprises cherchant à améliorer leur compétitivité et leur réactivité face aux exigences du marché. L'automatisation, l'intelligence artificielle et l'Internet des objets redéfinissent les processus industriels, rendant les échanges de données plus complexes mais aussi plus essentiels. Dans ce contexte, la capacité des entreprises à connecter efficacement leurs différents outils numériques détermine en grande partie leur performance et leur agilité face aux évolutions du secteur. Parmi ces défis, l'interopérabilité entre les systèmes de gestion et ceux pilotant la production constitue un point névralgique, influençant directement la productivité et la qualité des produits finis. Il constitue également un prérequis pour la mise en œuvre concrète des principes de l'Industrie 4.0, en posant les bases d'un écosystème industriel connecté et réactif.

En effet, les entreprises manufacturières doivent composer avec une diversité de solutions logicielles couvrant à la fois les aspects stratégiques, tactiques et opérationnels, tels que les progiciels de gestion intégrés, dits ERP (Enterprise Resource Planning) et les logiciels de pilotage de la production, dits MES (Manufacturing Execution System). Ces solutions incluent des systèmes de planification, de gestion des stocks, de suivi de production et de contrôle qualité, chacun répondant à des besoins spécifiques. Cependant, en fonction des fournisseurs, les architectures de ces solutions sont hétérogènes. Qui plus est, il n'existe pas de standard pour une communication fluide entre ces logiciels, ce qui engendre des inefficacités : duplication des données, erreurs de synchronisation, pertes d'informations et manque de visibilité sur l'ensemble du processus de fabrication. Cette fragmentation complexifie la prise de décision et nuit à l'optimisation des opérations, rendant plus difficile l'adaptation aux exigences du marché, telles que la personnalisation des produits, la réduction des délais de livraison et le respect des normes de conformité. Dès lors, une intégration efficace des systèmes informatiques devient un levier essentiel pour garantir une gestion industrielle cohérente et performante.

C'est dans ce contexte que ce mémoire s'intéresse aux enjeux liés au couplage entre les systèmes de gestion d'entreprise et ceux de gestion de la production, soit les ERP et les MES au niveau uniquement de la gamme de fabrication. L'objectif est de développer un cadre

d'interopérabilité, au niveau des gammes de fabrication, entre MES et ERP afin de favoriser leur intégration verticale en limitant les redondances et en améliorant la cohérence des informations échangées . En particulier, l'étude portera sur la modélisation d'une gamme de fabrication commune aux deux logiciels afin de faciliter la création et la mise à jour des gammes au sein de l'ERP et du MES en fonction des différents produits et processus de fabrication.

Pour répondre à cette problématique, une revue de littérature examinera les technologies de couplage entre MES et ERP, ainsi que les approches de modélisation des gammes de fabrication. Le chapitre 3 présentera la méthodologie adoptée. Le chapitre 4 portera sur le développement du cadre d'interopérabilité entre ERP et MES, avec la construction d'un modèle commun de gamme de fabrication, formalisé sous forme d'ontologie. La validation de ce modèle sera abordée au chapitre 5, à travers une vérification structurelle, sémantique et la réalisation de preuves de concept. Enfin, le chapitre 6 conclura sur les principales contributions de ce travail et les perspectives de recherche subséquentes.

CHAPITRE 2 REVUE DE LITTÉRATURE

2.1 Industrie 4.0

L'industrie 4.0 ou quatrième révolution industrielle représente une transformation profonde des processus de fabrication grâce à l'intégration des technologies numériques avancées. Le terme d'Industrie 4.0 a été introduit pour la première fois en 2011 en Allemagne lors de la foire d'Hanovre et désigne une initiative gouvernementale. Elle veut s'inscrire dans la continuité des précédentes révolutions industrielles, telles que décrites par ("Révolutions Industrielles", 2018) :

- Première révolution industrielle (fin du XVIIIe siècle) : Introduction de la mécanisation avec l'utilisation de la machine à vapeur ;
- Deuxième révolution industrielle (fin du XIXe siècle) : Adoption de l'électricité en autre et de nouveaux modèles d'organisation, permettant la production de masse ;
- Troisième révolution industrielle (milieu du XXe siècle) : Émergence de l'électronique et de l'informatique, conduisant à l'automatisation des processus de production ;

De nombreuses définitions officielles ont émergé depuis 2011. On peut retenir plus spécifiquement deux d'entre elles. La première de Danjou, Rivest et Pellerin (Danjou et al., 2017a) la définit comme " une stratégie qui s'appuie sur le numérique et la connectivité ubiquitaire, grâce à différentes technologies, afin de transformer les processus, les produits et les services par une prise de décision en temps réel et décentralisée permettant l'apparition de nouvelles capacités pour les systèmes, en coopération avec les humains, pour aller de la surveillance à l'autonomie." Dans la seconde (Vollmar, 2009) définit les bases de l'industrie 4.0 comme "Integration, communication and cooperation of all elements of the value chain vertically and horizontally across boundaries in real-time", ce que l'on pourrait traduire par : l'intégration, la communication et la coopération de tous les éléments de la chaîne de valeur, à la fois horizontale et verticale, au-delà des frontières, en temps réel.

Dans ce cadre, l'Industrie 4.0 est marquée par l'intégration de groupes de technologies numériques avancées, transformant les modèles de production traditionnels. Dix groupes technologiques clés forment alors son socle et sont décrits notamment par (Danjou et al., 2017b) :

- Les données massives (Big Data) ; Exploitation de vastes quantités de données issues de diverses sources pour optimiser la prise de décision, notamment grâce à des technologies comme le calcul haute performance et les plateformes analytiques ;
- L'intelligence artificielle ; Développement de systèmes capables de réaliser des tâches nécessitant des capacités cognitives avancées, comme l'apprentissage et la prise de

décision, en s'appuyant sur des réseaux de neurones et l'apprentissage automatique ;

- L'infonuagique (Cloud Computing) ; Accès à distance à des ressources informatiques partagées (stockage, logiciels, puissance de calcul), facilitant la gestion des infrastructures et l'échange de données dans les environnements industriels ;
- L'internet des objets (Internet of Things, IoT) ; Réseau interconnecté d'objets physiques et numériques permettant d'échanger des informations et d'agir en temps réel sur les processus industriels via des technologies comme les capteurs et les puces RFID ;
- Les systèmes cyberphysiques (Cyber-Physical Systems, CPS) ; Intégration de composants physiques et numériques capables d'échanger des données et d'interagir de manière autonome avec leur environnement, optimisant la surveillance et le contrôle des processus ;
- La cybersécurité ; Mise en place de protocoles et de technologies sécurisant les échanges de données et l'accès aux systèmes industriels pour prévenir les cyberattaques et garantir l'intégrité des opérations ;
- Les robots/machines autonomes ; Équipements industriels capables de fonctionner de manière indépendante, adaptative et collaborative, améliorant la flexibilité et la sécurité des interactions homme-machine ;
- La communication intermachines (Machine to Machine, M2M) ; Établissement de réseaux autonomes entre machines, distincts de l'IoT, permettant l'échange direct d'informations pour une coordination optimisée des opérations ;
- Les systèmes de simulations ; Création de répliques virtuelles de systèmes industriels (jumeaux numériques) pour tester et optimiser les processus avant leur mise en œuvre réelle, réduisant les temps d'ajustement et les erreurs ;
- La réalité augmentée ; Superposition d'informations numériques sur le monde réel afin d'assister les opérateurs en production, notamment pour l'assemblage, la maintenance ou la logistique ;

L'Industrie 4.0 repose ainsi sur un ensemble de groupes technologiques complémentaires qui transforment les processus de production. L'intégration de ces technologies permet non seulement d'améliorer l'efficacité opérationnelle, mais aussi d'optimiser la flexibilité et la réactivité des entreprises face aux exigences du marché. Cependant, pour exploiter pleinement le potentiel de cette transformation numérique, il est essentiel de disposer d'outils capables d'orchestrer et de structurer ces innovations au sein des systèmes industriels.

Dans ce contexte, les logiciels de gestion de production et/ou d'entreprise tels que les MES et ERP jouent un rôle fondamental dans la mise en œuvre des principes de l'Industrie 4.0. En effet, l'interconnexion entre ces systèmes et les nouvelles technologies permet d'utiliser le plein potentiel de ces dernières. Nous allons donc examiner en détail le rôle de ces systèmes

et leur intégration au sein des environnements industriels.

2.2 MES, ERP, leur rôle dans l'industrie 4.0

2.2.1 MES et ERP

Dans le cadre de l'industrie 4.0, les logiciels de gestion d'entreprise tels que les MES ou ERP jouent un rôle stratégique, car ils constituent des nœuds centraux de l'écosystème numérique industriel. Leur intégration et leur interopérabilité sont des conditions préalables à la mise en œuvre concrète des principes de l'Industrie 4.0, notamment l'automatisation avancée, la traçabilité complète et l'optimisation en temps réel des processus. En effet, ces deux logiciels permettent d'approcher l'excellence opérationnelle et de rester concurrentielles pour les entreprises (Shevchenko et al., 2022). L'article (Choi et Choi, 2019) observe ainsi une augmentation de la qualité des produits grâce à un MES ou un ERP spécifiquement pour les PME et ce, bien que ces dernières soient moins enclines à les adopter en raison des investissements financiers et en temps initiaux requis. La force de ces deux logiciels réside dans leur champ d'action et leur horizon de temps respectifs. D'un côté, les ERP permettent d'avoir une vue d'ensemble de la planification et de la gestion des ressources, opérant sur des horizons plus longs (planification tactique), tandis que les MES interviennent sur le court terme, au plus près du plancher de production, pour superviser et ajuster l'exécution des opérations en temps réel (Jaskó et al., 2020). Il convient alors d'examiner plus en détail leur rôle respectif. Un ERP est un système qui permet d'unifier et de gérer les processus clés d'une entreprise, tels que les finances, les ressources humaines, la production et la chaîne d'approvisionnement. Ces logiciels sont principalement conçus pour améliorer la communication et l'efficacité opérationnelle des processus d'affaires afin d'optimiser la prise de décision et de s'adapter rapidement aux évolutions du marché d'après (Romero et Vernadat, 2016). Les MES, quant à eux, sont plus proches du plancher de production. Leur rôle est de superviser et d'optimiser les processus de production en temps réel. Ils assurent alors la coordination entre les différentes opérations de fabrication afin de garantir une exécution efficace et la traçabilité complète des produits d'après (Shojaeinasab et al., 2022). Ces deux logiciels ont donc un rôle complémentaire à jouer. Leur intégration efficace nécessite alors un cadre structurant permettant d'assurer l'interopérabilité et l'intégrité des systèmes : c'est ce qu'on appelle l'intégration verticale.

C'est à cet effet que le standard ISA 95 (*Enterprise-control system integration. Part 1*, 2000) a été rédigé. Il définit une architecture hiérarchique visant à organiser les échanges de données entre les systèmes d'information d'entreprise et les systèmes de contrôle industriel. Ce modèle repose sur cinq niveaux distincts (figure 2.1), où l'ERP (niveau 4) gère les fonctions

de planification et de gestion des ressources, tandis que le MES (niveau 3) assure l'exécution des opérations et la coordination en temps réel des activités de production. Les niveaux inférieurs (0 à 2) englobent les équipements physiques, capteurs et automates industriels qui alimentent le MES en données essentielles au suivi des performances et à l'optimisation des processus.

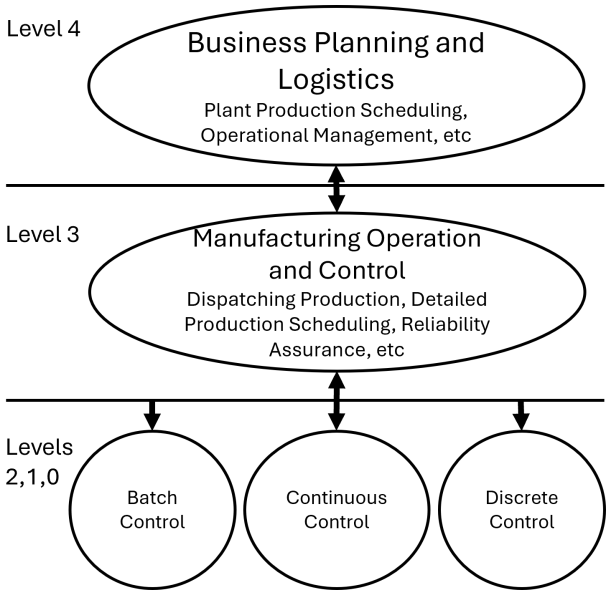

FIGURE 2.1 Schéma de la hiérarchie fonctionnelle, (*Enterprise-control system integration. Part 1*, 2000)

En plus de définir une hiérarchie des niveaux de gestion, la norme ISA-95 fournit également un modèle de données standardisé visant à structurer et harmoniser les échanges d'informations entre l'ERP et le MES. Ce modèle décrit les objets métier clés, tels que les ordres de fabrication, les ressources, les statuts de production et les nomenclatures, permettant ainsi une communication fluide et cohérente entre les différents systèmes. Cependant, si la norme ISA-95 pose un cadre pour l'intégration verticale entre MES et ERP (entre autres), différentes technologies ont été exploitées pour atteindre cette intégration dans les faits.

2.2.2 Les approches pour l'intégration verticale entre MES et ERP

L'intégration verticale des systèmes ERP et MES, ainsi que leur interopérabilité, est explorée par la norme ISA 95. Elle constitue même un levier stratégique pour renforcer la réactivité des entreprises face aux fluctuations de la demande et aux perturbations survenant en production selon (PANETTO et MOLINA, 2008). Toutefois, la consolidation des informations entre ces deux systèmes reste, dans les faits, complexe, souvent partielle ou particulièrement laborieuse,

entraînant des doublons dans la saisie des données, des incohérences et une diminution de l'efficacité dans la gestion des informations manufacturières, d'après (Shevchenko et al., 2022). Pour faciliter ce couplage entre les ERP et les MES, différentes approches ont été exploitées :

— **L'Enterprise Application Integration** (EAI) désigne un ensemble de techniques et d'outils permettant de faire communiquer des applications hétérogènes au sein d'un système d'information, sans qu'elles aient été conçues initialement pour interagir. Bien que le concept ait émergé aux États-Unis dans les années 1990, ses premières applications remontent aux années 1970 en France, notamment dans le secteur bancaire. L'EAI vise principalement à assurer l'échange fluide de données entre des logiciels d'entreprises comme les ERP, CRM ou SCM, en s'appuyant sur des mécanismes de transformation d'événements et de règles, d'après Manouvrier et Ménard (Manouvrier et Ménard, 2008). Contrairement aux solutions d'architecture interne d'une application, l'EAI se concentre sur l'interconnexion de systèmes distincts, ce qui en fait un élément clé pour l'intégration entre un ERP et un MES ;

— Parallèlement à l'EAI, la technologie **Extract, Transform, Load** (ETL) s'est développée pour répondre aux besoins d'agrégation et d'analyse des données industrielles. Il repose sur trois étapes essentielles : l'extraction des données depuis diverses sources, leur transformation selon les exigences métier et leur chargement dans une base cible (Florea et al., 2015). Cette approche permet d'uniformiser et de structurer les données, facilitant ainsi leur exploitation. Putro et al. (Putro et al., 2018) soulignent son rôle dans la mise en place d'entrepôts de données industriels, optimisant la traçabilité et la gestion des processus de production. De leur côté, Jiantao et al. (Jiantao et al., 2013) mettent en avant l'apport des techniques avancées, telles que les réseaux neuronaux, pour améliorer la précision du traitement des données. Toutefois, malgré ces avantages, l'ETL présente certaines limites. Son fonctionnement en mode batch peut être un frein pour les environnements nécessitant des mises à jour en temps réel. De plus, son déploiement et sa maintenance exigent une expertise technique avancée, notamment pour l'adaptation aux évolutions des structures de données et la gestion des volumes croissants d'informations ;

— Avec l'essor des **architectures orientées services** (Service-Oriented Architecture – SOA) dans les années 2000, une approche plus modulaire et flexible a émergé pour faciliter l'intégration des systèmes ERP et MES. Contrairement aux solutions EAI et ETL, qui reposent sur des infrastructures centralisées, SOA privilégie une communication décentralisée basée sur des services web exposés via des protocoles standards tels que SOAP et REST. L'approche SOA permet une intégration plus agile et évolu-

tive, chaque système exposant des services accessibles par d'autres applications sans dépendance forte entre elles. L'utilisation des API Web (Application Programming Interfaces) facilite l'interconnexion entre l'ERP et le MES, permettant, par exemple, de récupérer en temps réel les ordres de fabrication ou les données de suivi de production sans nécessiter un middleware intermédiaire comme montré par Beric et al (Beric et al., 2020). Cette approche s'est imposée comme un standard pour les nouvelles générations de logiciels industriels, bien qu'elle requière une structuration rigoureuse des données échangées et une gestion efficace des performances réseau ;

— **L'Enterprise Service Bus** (ESB), apparu au cours des années 2000-2015, s'inscrit dans l'évolution des solutions d'intégration en reprenant et en améliorant les concepts de l'EAI et du SOA. Cette technologie introduit des capacités avancées de médiation et d'orchestration des flux d'information. Wang et al. (Wang et al., 2010) le définissent comme un cadre facilitant la communication entre systèmes hétérogènes, offrant notamment des services de transformation des messages, de routage intelligent et de gestion des protocoles. Son architecture permet ainsi de standardiser les échanges tout en réduisant la complexité des intégrations point-à-point. Pour sa part, Dai (Dai, 2011) met en évidence l'intérêt de l'ESB dans le domaine de la gestion énergétique, en soulignant son rôle dans l'amélioration de l'adaptabilité et de la fluidité des interactions entre applications réparties. Néanmoins, bien que l'ESB présente des avantages significatifs, certaines limites doivent être prises en compte. Sa mise en œuvre peut être complexe et requérir une expertise technique poussée, tant pour son installation que pour sa maintenance. En outre, les investissements nécessaires en infrastructures et en formation du personnel peuvent représenter un frein à son adoption. Enfin, si cette technologie favorise l'interopérabilité, une configuration inadéquate ou une surcharge du bus peut entraîner des ralentissements, compromettant ainsi les performances globales du système. Par conséquent, son intégration doit être soigneusement planifiée afin de maximiser ses bénéfices tout en minimisant ses contraintes ;

— L'essor du **cloud computing** au cours des années 2010 a introduit de nouvelles possibilités d'intégration entre ERP et MES, notamment via des plateformes d'intégration en tant que service (Integration Platform as a Service – iPaaS). Ces solutions permettent de connecter des applications locales et cloud sans nécessiter d'infrastructure lourde sur site. Les solutions d'intégration basées sur le cloud offrent plusieurs avantages, notamment une réduction des coûts d'infrastructure, une scalabilité accrue et une mise en œuvre plus rapide des échanges de données. Elles sont particulièrement adaptées aux PME cherchant à simplifier l'intégration de leurs systèmes tout en bénéficiant d'une flexibilité accrue selon les travaux de (Morariu et Borangiu, 2012).

Toutefois, ces solutions impliquent des défis en matière de sécurité des données et de dépendance vis-à-vis des fournisseurs de services cloud ;

Cette section nous a permis d'explorer les différentes façons de réaliser un cadre d'interopérabilité entre MES et ERP. Cependant, nous souhaitons, dans cette recherche, établir un modèle de gamme de fabrication commun à ces deux logiciels pour faciliter ce lien. C'est pourquoi nous verrons dans la prochaine section les différentes bases de données envisageables pour une gamme.

2.3 Choix d'architecture de modèle de gamme de fabrication

Pour construire une plateforme de couplage des logiciels ERP et MES sur les gammes de fabrication, il est nécessaire de construire un modèle commun de gamme de fabrication où se retrouvent les informations issues du MES et de l'ERP et les informations dont ils ont besoin pour fonctionner. Le choix du modèle unifié de données est donc crucial et est conditionné par les contraintes spécifiques à la production industrielle. Or de nombreux modèles existent dans la littérature, on peut ainsi citer les modèles hiérarchiques, en réseau, relationnels, orientés objet, NoSQL et ontologiques. Afin de comparer les modèles et sélectionner le plus pertinent à notre usage, nous avons construit un tableau d'évaluation multicritère : nous avons, dans un premier temps, extrait de la littérature les critères d'évaluation de modèle les plus pertinents et dans un second temps, nous avons évalué et noté chaque modèle selon ces critères afin de mettre en évidence le modèle le plus judicieux pour notre plateforme.

2.3.1 Sélection des critères d'évaluation

Dans le cadre de la sélection d'un modèle de base de données adapté à la centralisation des informations entre les systèmes ERP et MES, il est crucial d'établir des critères de comparaison et donc de sélection clairs et rigoureux. La diversité des six (6) modèles de bases de données étudiés, c'est-à-dire les modèles : hiérarchique, en réseau, relationnel, orienté objet, NoSQL et ontologique, nécessite une analyse approfondie des caractéristiques qui les différencient. Cette comparaison permettra d'évaluer la pertinence de leur application dans le cadre de la mise en commun des gammes de fabrication des deux systèmes de gestion d'entreprise.

Pour définir les critères comparatifs, nous nous appuyons sur la littérature scientifique afin d'identifier les attributs les plus pertinents utilisés en recherche pour chaque modèle. En effet, de nombreuses comparaisons entre ces modèles existent au sein de la littérature, fournissant par la même occasion un certain nombre de critères que nous allons étudier et, s'ils sont pertinents, réutiliser. Nous pouvons ainsi trouver des articles comparant le NoSQL et les bases de données relationnelles (Phiri et Kunda, 2017), le modèle relationnel à celui orienté objet

(Suri et Sharma, 2011), comparant ontologie et base de données (Sir et al., 2015).
Parmi les très nombreux critères existants, leur sélection s'est basée sur leur caractère déterminant pour le contexte spécifique d'une plateforme de couplage ERP-MES et sur la capacité des critères à pouvoir évaluer les 6 modèles sans exception.
Ainsi cette étape d'analyse des critères assure que l'évaluation des six modèles offre, dans les faits, le modèle le plus pertinent pour le contexte étudié.

2.3.1.1 Extraction des critères d'évaluation

Dans un premier temps, nous allons extraire de la littérature et donc des articles, les critères qui ont été sélectionnés pour comparer les différents modèles. Dans un second temps, nous pourrons sélectionner ou non ces différents critères en fonction de notre étude spécifique.

Dans l'article, Comparative Study of NoSQL and Relational Database (Phiri et Kunda, 2017), onze (11) critères de comparaison entre modèle de base de données relationnel et NoSQL sont proposés. Énumérons-les accompagnés d'une courte définition. La **variété** (« Variety ») des solutions, c'est-à-dire s'il existe ou non des plateformes OpenSource et fermées pour la base de données. La **scalabilité** (« Scalability ») ou la capacité d'un système à supporter une augmentation du volume de données ou d'utilisateurs, en utilisant des ressources matérielles plus puissantes ou en répartissant la charge sur plusieurs serveurs. Le **coût** (« Cost ») lié à l'utilisation, la maintenance ou la mise à niveau d'une base de données. »). Le **volume de données** (« Volume of Data ») ou la capacité du système à stocker et traiter une large quantité de données. La **disponibilité** (« Availibility ») ou la capacité de rester opérationnel et accessible pour un certain nombre d'utilisateurs malgré des pannes. La **performance face aux requêtes** (« Performance ») c'est-à-dire la rapidité et l'efficacité de traitement de requêtes plus ou moins complexes. La **complexité** (« Complexity ») fait ici référence au degré de difficulté pour mettre en place et manipuler des données plus ou moins complexes. Le **langage de requête** (« Query Language ») utilisé pour interagir avec le système. La **cohérence des données** ou **l'intégrité** (« Consistency ») c'est-à-dire la capacité de garantir que toutes les transactions aboutissent à un état cohérent et uniforme pour le système. Enfin, la **sécurité** (« Security ») qui garantit la confidentialité et l'intégrité des données et leur accès.

Ensuite avec l'article A comparative study between the performance of relational et object oriented database in data warehousing (Suri et Sharma, 2011), nous pouvons extraire 6 critères (en excluant ceux très spécifiques au modèle de données orienté objet). De la même manière que précédemment, énumérons-les accompagnés d'une courte définition. Tout

d'abord, la capacité de gérer des **types de données complexes** ("different types of data types"), c'est-à-dire pouvoir manipuler des données multimédia, des objets hiérarchiques ou des structures définies par l'utilisateur. La **flexibilité de développement** ("Composite Object and relationship") ou le niveau de souplesse offert par le modèle dans les processus de développement. L'**extensibilité** ("Composite Object and Relationship" et "no impedance mismatch ") soit la capacité de la base de données à être augmentée ou modifiée facilement sans perturber les fonctionnalités existantes, permettant d'ajouter de nouveaux types de données ou des évolutions fonctionnelles. L'**expressivité** (avec "Class Hierarchy", et "One Data Model") ou la facilité avec laquelle un modèle de données peut représenter des relations hiérarchiques et complexes, permettant une représentation intuitive et complète des entités et de leurs interactions. Le **langage de requête** ("Query language") déjà défini au paragraphe précédent. Finalement, la **complexité** (via "No Primary Keys"), qui a aussi déjà été explicité précédemment.

L'article suivant intitulé, Ontology versus Database (Sir et al., 2015) permet de mettre en avant 8 critères de comparaison pour des bases de données, c'est-à-dire non spécifiques aux ontologies. Énumérons-les en définissant ceux que nous n'avons pas encore rencontrés. En premier lieu, la flexibilité de développement (via "Data dependency", "Reduced application development time" et "To separate the domain knowledge from the operational knowledge") définie au paragraphe précédent. La performance avec des requêtes complexes (via "Efficient data access") également traitée par le premier article. La cohérence ou intégrité des données (via "Data integrity") précédemment définie. La sécurité (via "security") précédemment définie également, l'administration des données (via "Data administration"), c'est-à-dire l'ensemble des pratiques et procédures permettant la gestion, la sécurisation et la supervision des données d'une base de données, incluant leur mise à jour, leur sauvegarde et leur archivage. L'accès concurrent et récupération après un crash ("concurrent access and crash recovery") ou les mécanismes permettant à plusieurs utilisateurs d'accéder et de modifier simultanément les données dans une base de données, tout en assurant la cohérence et la stabilité des informations. L'interopérabilité (via "To share common understanding of the information structure between people or software"), capacité donc à échanger des informations avec d'autres systèmes en partageant une compréhension commune de la structure. Finalement, l'extensibilité ("To enable reuse of the domain knowledge") que nous avons déjà définie.

2.3.1.2 Critères sélectionnés

Finalement, 16 critères de comparaison ont pu être extraits dans les articles (Phiri et Kunda, 2017), (Suri et Sharma, 2011), (Sir et al., 2015). Discutons désormais de leur pertinence

vis-à-vis de notre étude et sélectionnons les plus utiles :

- La variété est un critère à conserver afin de conserver une autonomie et une indépendance face aux logiciels utilisés dans le cadre de notre étude ;
- La scalabilité est à conserver car la base de données devra évoluer au même rythme que l'entreprise, surtout pour des solutions MES collectant des données en temps réel ;
- Le coût n'est pas conservé, dans notre contexte académique, bien que le coût global soit important. L'étude se concentre sur l'aspect technique et les bénéfices à long terme liés à la fiabilité et à la réduction des erreurs ;
- Le volume de données est à conserver en raison de la quantité d'informations à gérer dans les gammes de fabrication, notamment dans des entreprises qui collectent des données en temps réel pour le MES et des données stockées à des fins analytiques dans l'ERP ;
- La disponibilité n'est pas conservée. Ce critère est souvent plus critique dans les environnements à haute disponibilité comme les plateformes en ligne ou les systèmes critiques en temps réel. De plus, les systèmes ERP et MES ont également leurs propres solutions de reprise en cas de panne ;
- La performance face aux requêtes est à conserver pour garantir que les données circulent rapidement entre les systèmes. Les retards dans la transmission de données pourraient compromettre la mise à jour en temps réel et l'efficacité de la production ;
- La complexité est à conserver pour s'assurer que le développement, la maintenance et la gestion n'entraînent des coûts de formation élevés ou un temps excessif ;
- Le langage de requête n'est pas conservé. Nous cherchons ici à démontrer la faisabilité et la pertinence de notre solutions sans contraintes sur la technique ;
- L'intégrité des données est à conserver. En effet, assurer fiabilité et la synchronisation des informations utilisées par les systèmes ERP et MES lors de la configuration ou de leur mise à jour est plus que vital ;
- La sécurité n'est pas conservée. Elle est gérée par les systèmes ERP et MES eux-mêmes ;
- La gestion de types de données complexes est à conserver. En effet, dans le cadre d'un couplage de gamme de fabrication, les données échangées peuvent inclure des informations multimédia, des structures hiérarchiques, ou d'autres formats complexes ;
- La flexibilité de développement est à conserver pour permettre de personnaliser les processus de développement et d'adapter les fonctionnalités aux spécificités des systèmes ERP et MES en simplifiant la configuration et la gestion ;
- L'extensibilité est à conserver en raison de l'évolution continue des besoins et des technologies de fabrication. Il s'agit d'un critère essentiel pour permettre de modifier

ou d'ajouter des fonctionnalités sans perturber l'intégrité des données existantes ;
— L'expressivité est à conserver dans le cadre de gammes de fabrication. En effet, la capacité de représenter des concepts complexes avec richesse et clarté sémantique permet de structurer les données de manière plus intuitive pour les utilisateurs et pour l'organisation des informations ;
— L'administration des données n'est pas conservée. Elle est, en partie, assurée par d'autres critères (intégrité) et chaque système ERP ou MES dispose déjà de pratiques robustes de gestion des données ;
— L'accès concurrent et récupération après un crash n'est pas conservé car le modèle centralisé ne vise pas des interactions massives simultanées, et les ERP et MES incluent déjà des fonctions de reprise après incident ;
— L'interopérabilité est à conserver, il s'agit d'un des critères les plus essentiels de notre étude. En effet, nous avons besoin de pouvoir intégrer facilement notre solution à la fois aux systèmes ERP et aux systèmes MES ;

Finalement 12 critères de comparaison sont conservés. Il s'agit de la variété, la scalabilité, le volume de données, la performance face aux requêtes, la complexité, l'expressivité, la gestion de types de données, la flexibilité, l'extensibilité, l'intégrité des données et l'interopérabilité. Ces trois derniers critères sont par ailleurs les plus décisifs dans le choix de notre modèle pour le couplage entre ERP et MES au niveau des gammes de fabrication. C'est sur la base de ces 12 critères que nous baserons l'évaluation des modèles de données qui suit.

2.3.2 Évaluation des modèles de données

Après avoir établi les critères de sélection qui guideront notre analyse comparative, nous proposons une évaluation multicritère des différents modèles de bases de données couramment utilisés dans les systèmes d'information. Les modèles évalués sont les bases de données hiérarchiques, en réseau, relationnelles, orientées objet, NoSQL, ainsi que les systèmes basés sur des ontologies. Cette évaluation des modèles repose sur une analyse de la littérature à la fois spécifique de chaque modèle mais également comparative entre différents modèles. Pour chaque modèle, une note comprise entre – et ++ (sans note nulle) a été attribuée afin de faire ressortir le ou les modèles les plus pertinents. Nous avons donc obtenu le tableau 2.1, un tableau avec une justification pour chaque critère et chaque modèle basé sur la littérature est en Annexe A. À la suite de cette évaluation multicritère, deux modèles de base de données se détachent, le modèle ontologique avec 17 (dix-sept) points et le modèle orienté-objet avec 16 (seize) points. Ces deux modèles ayant des notes relativement proches, évaluons-les sur les critères les plus décisifs pour notre étude : l'extensibilité, l'intégrité des données et l'interopérabilité, d'après l'extraction des critères précédente.

Tableau 2.1 Comparaison des systèmes de gestion de données

Critères	**Hiérarchique**	**En réseau**	**Relationnel**	**Orienté objet**	**NoSQL**	**Ontologie**
Variété des solutions	-	+	+	+	+	+
Scalabilité	+	+	+	+	++	+
Volume de données	+	++	+	++	++	+
Performance de requêtes	-	++	+	++	++	++
Complexité	++	-	-	- -	-	- -
Intégrité de données	+	+	++	+	- -	++
Sécurité	+	+	++	+	-	+
Type de données complexes	+	+	+	++	++	++
Maturité du marché / Maintenance	++	+	++	+	+	+
Extensibilité	-	+	+	+	++	++
Flexibilité de développe-ment	- -	+	+	++	++	++
Interopérabilité	- -	+	+	+	+	++
Expressivité	- -	+	+	++	+	++
Total	0	13	14	16	12	17

D'après l'analyse multicritère, le modèle de données ontologiques est supérieur au modèle orienté objet de 1 point sur chacun des trois critères. C'est pourquoi le choix s'est porté sur le modèle ontologique. Ce type de modèle offre une grande flexibilité pour représenter et faire évoluer la structure des données sans remise en cause complète du schéma initial. Il permet également une centralisation de l'information dans un cadre conceptuel unifié, favorisant la cohérence et la réduction des redondances. De plus, les ontologies sont particulièrement adaptées aux requêtes rapides et complexes grâce à leur capacité à raisonner sur les relations sémantiques entre concepts, ce qui facilite l'extraction d'informations pertinentes même à partir de données hétérogènes. Enfin, elles constituent un socle naturel pour l'interopérabilité entre systèmes, en fournissant un langage commun et formel de description des données.
Cette section a permis de mettre en évidence que le modèle ontologique semble être un choix judicieux dans notre étude, et c'est vers celui-ci que l'on s'orientera au chapitre 4. Dans la

section suivante, nous allons regarder les différentes façons dont le modèle ontologique a été exploité dans le cas des gammes de fabrication dans l'industrie.

2.4 Les modèles ontologiques dans le manufacturier

Un des défis de l'industrie manufacturière est de travailler avec des données d'une grande diversité et complexité. La littérature s'est donc intéressée à des outils de traitement des données qui permettent de représenter les données et leurs liens. L'ontologie apparaît dans ce contexte comme un outil privilégié, c'est un modèle de données qui permet de représenter un ensemble de concepts d'un domaine ainsi que l'ensemble des relations liant ces concepts. Trois grands types d'ontologies existent, qui vont de la plus globale à la plus fine : les ontologies top-level ou globales, les ontologies de domaines et les ontologies d'application (Guarino, 1998).

— Les ontologies globales visent à modéliser les concepts les plus généraux, indépendamment d'un domaine particulier, tels que *objet*,*événement*, *temps espace*, ou *relation causale*. Elles constituent une base conceptuelle commune sur laquelle peuvent s'appuyer d'autres types d'ontologies plus spécifiques ;
— Les ontologies de domaine se concentrent sur la description des concepts propres à un domaine particulier de connaissance ou d'activité, par exemple la médecine, la mécanique ou la logistique. Elles permettent de partager une terminologie normalisée au sein d'une communauté professionnelle ;
— Les ontologies d'application, enfin, sont conçues pour des cas d'usage spécifiques et opérationnels. Elles décrivent des entités, des propriétés et des relations adaptées à une tâche précise, souvent avec un niveau de granularité plus fin que celui des ontologies de domaine ;

Cette typologie met en lumière l'importance d'adapter le niveau d'abstraction de l'ontologie à son objectif.

Comme dit en section précédente, les ontologies sont de plus en plus utilisées dans le domaine manufacturier, bien que cet outil reste très académique. Regardons alors les différents projets de développement d'ontologies dans le domaine manufacturier des gammes de fabrication.

— L'équipe de (Ameri et Dutta, 2006) propose une ontologie dédiée à la structuration et à la formalisation des connaissances manufacturières, avec pour objectif d'améliorer l'interopérabilité entre les systèmes industriels. Leur approche s'appuie sur la norme ISO 10303-224 (STEP AP224), qui définit un cadre standardisé pour la représentation des processus de fabrication et des ressources associées. L'ontologie développée modé-

lise les entités clés du domaine, telles que les opérations, les équipements, les outils et les matériaux, ainsi que leurs relations, permettant ainsi une description cohérente et exploitable des gammes de fabrication. Ce modèle favorise l'intégration et le partage des informations entre les différentes applications industrielles, tout en facilitant la réutilisation des connaissances manufacturières. En structurant ces données de manière formelle, l'ontologie contribue à la standardisation des échanges d'informations et ouvre la voie à une automatisation accrue des processus décisionnels, en fournissant une base exploitable par des outils d'aide à la décision et d'optimisation des opérations industrielles ;

— L'équipe de (Saha et al., 2023) propose une ontologie centrale pour la modélisation des opérations manufacturières et de leur séquencement, visant à améliorer la transmission des connaissances en ingénierie de fabrication. L'approche ontologique permet de structurer les relations entre différentes opérations et de formaliser les séquences de production à l'aide du Web Ontology Language (OWL) et de règles définies en Semantic Web Rule Language (SWRL). La validation de ce modèle dans un environnement industriel, notamment au sein d'une grande entreprise aéronautique, démontre son applicabilité pour la gestion des connaissances en fabrication. Contrairement aux modèles existants, cette ontologie intègre une granularité fine des opérations, permettant ainsi une meilleure prise en charge des scénarios de planification manufacturière et facilitant la prise de décision des ingénieurs lors de la planification de nouveaux produits ;

— L'équipe de (Nagy et al., 2021) met en avant l'utilisation d'une ontologie pour analyser les processus manufacturiers à travers une étude de cas sur la production de faisceaux de câblage. Leur approche repose sur l'interconnexion des données à l'aide de graphes de connaissances et l'analyse en réseau multicouche. L'étude illustre comment l'ontologie permet de modéliser et d'optimiser la gestion des processus d'assemblage, en identifiant les séquences critiques et en facilitant la prise de décision dans un contexte de fabrication complexe. L'un des principaux apports de cette recherche est la mise en évidence du rôle des ontologies dans la structuration des données de production et leur exploitation pour améliorer l'efficacité des processus industriels, notamment en matière de planification et d'ordonnancement ;

Ainsi, les ontologies sont utilisées en milieu manufacturier et permettent de définir des processus ou des gammes de fabrication en identifiant les concepts clés et leurs relations.

2.5 Revue critique

L'analyse de la littérature réalisée dans les sections précédentes a mis en lumière les transformations majeures induites par l'Industrie 4.0 depuis les années 2010 dans les environnements manufacturiers. L'émergence de cette intégration des technologies numériques a permis d'améliorer la flexibilité, la réactivité et l'efficacité des processus industriels. Cependant, cette mutation s'accompagne de défis significatifs en matière d'interopérabilité et d'interconnexion entre les technologies désormais omniprésentes dans les entreprises industrielles et notamment entre les systèmes d'information et entre les logiciels ERP et MES, dont les fonctions sont à la fois complémentaires et interdépendantes.

L'intégration fluide de ces systèmes est donc un enjeu central, largement exploré par la littérature à travers différentes technologies de couplage telles que l'EAI, l'ETL, le SOA, l'ESB ou encore les plateformes cloud. Toutefois, bien que ces solutions permettent d'assurer un certain niveau de communication entre ERP et MES, elles présentent des limites importantes. Elles traitent principalement les aspects techniques du transfert de données, mais ne répondent pas toujours aux problématiques sémantiques : chaque système possède sa propre manière de structurer les informations, rendant difficile une compréhension partagée sans redondance ni ambiguïté. C'est dans ce contexte qu'un modèle de données unifié devient indispensable pour assurer une synchronisation cohérente, notamment dans le cadre des gammes de fabrication, qui sont des objets complexes manipulés par les deux systèmes.

Dans cette optique, notre revue a comparé différents types de modèles de bases de données susceptibles de centraliser l'information issue des deux systèmes. L'évaluation multicritère a permis d'identifier le modèle ontologique comme le plus pertinent, notamment en raison de ses atouts en matière d'extensibilité, d'intégrité des données et d'interopérabilité – trois critères essentiels dans notre contexte d'étude. Contrairement à d'autres modèles comme le relationnel ou le NoSQL, le modèle ontologique se distingue par sa capacité à représenter de manière formelle et explicite les relations complexes entre concepts métier, ce qui en fait un excellent candidat pour structurer les gammes de fabrication de manière unifiée et compréhensible par les deux systèmes.

Nous avons ensuite analysé les modèles ontologiques appliqués à l'industrie manufacturière et plus précisément aux gammes de fabrication. Toutefois, ces modèles présentent certaines limites qui rendent leur application à notre problématique difficile. En effet, ils sont souvent conçus pour des cas d'usage spécifiques, adaptés à une industrie, un procédé de fabrication

ou un logiciel particulier. Par ailleurs, leur objectif premier n'est pas de faciliter le couplage entre MES et ERP, ce qui limite leur portée dans un contexte d'intégration globale. Les recherches existantes se concentrent avant tout sur la structuration des concepts et des relations au sein des ontologies, sans démontrer concrètement leur efficacité pour résoudre les problématiques d'intégration et de synchronisation des données entre systèmes industriels.

Ainsi, malgré les avancées réalisées dans ce domaine, aucun modèle de gamme de fabrication universel ne permet aujourd'hui de résoudre pleinement les problèmes d'interopérabilité entre MES et ERP. Il apparaît donc nécessaire de développer une approche innovante qui prenne en compte les spécificités des gammes de fabrication tout en garantissant une intégration cohérente et efficace entre ces deux systèmes fondamentaux de la gestion industrielle.

2.6 Conclusion

Les précédentes sections ont montré l'importance croissante de l'interopérabilité dans l'industrie manufacturière, en particulier dans le cadre de l'intégration des systèmes MES et ERP. Bien que plusieurs approches aient été développées pour faciliter cette intégration, elles présentent encore des limites, notamment en matière de flexibilité et de standardisation. L'examen des bases de données et des modèles ontologiques existants a révélé que, malgré leur fort potentiel, les solutions actuelles restent souvent spécifiques à un contexte particulier et ne permettent pas une communication fluide et généralisable entre les deux systèmes.

Face à ces constats, notre recherche propose le développement d'une solution d'interopérabilité basée sur un modèle ontologique de gamme de fabrication, capable de structurer et de relier efficacement les données des gammes de fabrication entre un MES et un ERP. Le prochain chapitre détaillera la méthodologie de conception de ce modèle puis sa mise en œuvre.

CHAPITRE 3 MÉTHODOLOGIE

Le chapitre précédent a mis en évidence l'importance de l'interopérabilité entre les systèmes de gestion d'entreprise dans le contexte de l'industrie 4.0. Il a également permis de constater l'absence d'un modèle de gamme de fabrication commun permettant de faciliter les échanges entre un ERP et un MES. Le présent chapitre décrit la méthodologie suivie pour concevoir une solution de couplage au niveau des gammes de fabrication, répondant aux besoins identifiés dans la revue de littérature. Pour ce faire, nous commencerons par préciser les objectifs et le cadre de notre recherche. Nous détaillerons ensuite la démarche suivie, qui comprend la construction d'un modèle commun de gamme et sa formalisation sous forme d'ontologie. Enfin, nous présenterons le mode de validation et d'évaluation de l'ontologie développée.

3.1 Objectif de recherche

La revue de littérature a permis de montrer à la fois le besoin et le manque d'interopérabilité entre les logiciels MES et ERP. Ce manque s'exprime de deux façons ; un manque d'alignement des concepts entre ces deux logiciels (M1) et un manque de communication entre ces deux logiciels (M2). D'autre part, elle a a aussi montré le fort potentiel des modèles ontologiques malgré leur utilisation encore limitée. Nous traiterons ainsi la problématique suivante : **Développer un modèle d'interopérabilité entre MES et ERP au niveau des gammes de fabrication**.

Pour ce faire, deux sous-objectifs se distinguent :

- Sous-objectif 1 : Développer un cadre d'interopérabilité MES/ERP qui répondra au M1 en permettant l'alignement de leurs concepts. ; c'est-à-dire développer un modèle de gamme de fabrication commun aux MES et ERP ;
- Sous-objectif 2 : Développer le support technique de l'interopérabilité entre MES et ERP pour répondre à M2 en favorisant la communication entre MES et ERP ; c'est-à-dire structurer le modèle commun de gamme sous forme d'ontologie ;

Pour répondre à cette problématique, nous allons donc nous intéresser à la gamme de fabrication et à la manière dont elle est définie dans chacun des deux logiciels. Une gamme de fabrication (ou *manufacturing process plan* ou *routing*) désigne un ensemble structuré d'opérations techniques, organisées selon un ordre logique, et visant à transformer un produit à travers plusieurs étapes de production. La norme ISA-95 partie 2 (*ANSI/ISA–95.00.02*, 2001) définit la gamme (appelée *Product production rule*) comme un regroupement structuré d'opérations de fabrication associées à des ressources, des entrées et des sorties. Les opérations sont

associées à des ressources (machines, opérateurs, documents), à des composants ou matières, ainsi qu'à d'éventuelles caractéristiques d'inspection pour la qualité.

Certains éléments souvent liés à une gamme dans les logiciels industriels seront alors exclus du périmètre de notre modélisation, car ils ne relèvent pas de la gamme mais relèvent d'autres éléments fonctionnels des logiciels MES ou ERP. Il s'agit par exemple : des données économiques (valorisation, rentabilité), de la gestion du personnel détaillée (affectations, compétences, disponibilités), de l'ordonnancement dynamique en temps réel. Bien sûr, certains éléments permettent de lier, grâce aux gammes, certains de ces éléments à d'autres modules de l'ERP ou du MES, ils seront donc conservés afin de maintenir les fonctionnalités des deux logiciels.

3.2 Méthodologie de recherche

Notre méthodologie de recherche doit nous permettre de répondre à la problématique globale de ce mémoire et donc de remplir les deux sous-objectifs cités précédemment en 3.1. Nous construirons ainsi dans un premier temps, un modèle de gamme de fabrication commun aux MES et aux ERP. Dans un deuxième temps, nous développerons l'ontologie associée à ce modèle, avant de la valider via expérimentation. Cette démarche générale de recherche est détaillée à la figure 3.1.

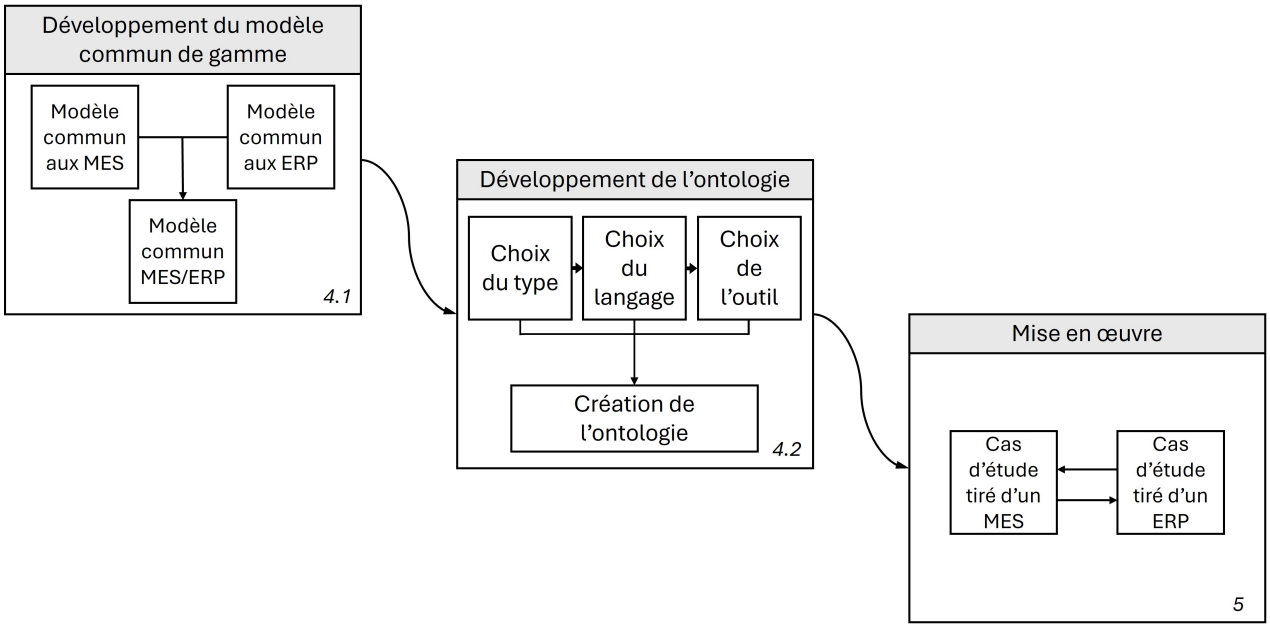

FIGURE 3.1 Schéma de la méthodologie globale

3.2.1 Développement du modèle commun de gamme

La première étape de notre méthodologie consistera à définir l'ensemble des champs de données qui devront figurer dans le modèle commun entre les systèmes ERP et MES. L'objectif est de concevoir un modèle de gamme de fabrication extensible, capable de répondre aux besoins à la fois en matière d'accès à l'information et de saisie de données, tant du côté des logiciels MES que des ERP. Afin d'atteindre cet objectif, nous structurerons notre démarche en trois phases distinctes successives. Dans un premier temps, un modèle de gamme orienté MES sera construit à partir de la norme ISA-95 partie 2 (*ANSI/ISA–95.00.02*, 2001), en croisant ses recommandations avec l'étude de deux logiciels MES utilisés dans le secteur manufacturier. Ensuite, un second modèle, cette fois orienté ERP, sera élaboré à partir de l'analyse de la structure de données de quatre ERP distincts. Enfin, ces deux modèles seront mis en commun afin d'aboutir à une structure commune, intégrant les champs de données essentiels pour assurer l'interopérabilité et le bon fonctionnement des deux types de systèmes.

3.2.1.1 Modèle commun aux MES

Pour la définition du modèle commun MES, nous nous appuierons en premier lieu sur la norme ISA-95 (notamment la partie 2 (*ANSI/ISA–95.00.02*, 2001)), qui fournit un cadre structurant et reconnu pour la modélisation des données de production. Ce référentiel sera complété par l'analyse de deux logiciels MES actuellement utilisés dans l'industrie manufacturière : Sepasoft et Smart Portal.

Concernant Sepasoft, nous nous appuierons sur la documentation très complète présente en libre accès ("Track and Trace Configuration", s. d.). Pour Smart Portal, logiciel moins documenté publiquement, l'analyse se fondera principalement sur des captures d'écran illustrant les champs de données minimaux nécessaires à la définition d'une gamme de fabrication.

La diversité des méthodes d'acquisition de données pourra impliquer une granularité inégale des champs de données récoltés. Cependant, les données recueillies au sujet de la norme ISA-95 seront exhaustives, celles issues de la documentation officielle de Sepasoft également, et les champs de données issus de Smart Portal correspondront à un modèle de gamme minimal. Ainsi, cela ne devrait pas empêcher de construire un modèle de gamme de fabrication compatible avec la majorité des MES.

Afin de construire un modèle générique, respectant à la fois les exigences de la norme ISA-95 et les spécificités des deux logiciels étudiés, tout en gardant une capacité d'extension vers d'autres MES, la démarche suivra les étapes suivantes :

1. **Élimination des champs de données spécifiques** au fonctionnement interne d'un

MES : C'est-à-dire les champs de données propres au fonctionnement interne d'un seul MES, sans valeur ajoutée pour la description, l'exécution, ou l'interopérabilité de la gamme, seront exclus.

2. **Harmonisation des champs de données similaires** : Les champs de données présentant une sémantique ou une fonction équivalente sont regroupés ou alignés, qu'il s'agisse d'une correspondance directe ou de différences de terminologie ou de typage.
3. **Évaluation des champs de données spécifiques** : Les champs de données présents dans un seul des deux MES font l'objet d'une évaluation en fonction de leur pertinence. Certains ont été retenus s'ils enrichissaient la description de la gamme, tandis que d'autres ont été exclus s'ils apparaissaient comme redondants, marginaux, ou en dehors du périmètre de cette étude.

3.2.1.2 Modèle commun aux ERP

Pour construire le modèle commun aux ERP, quatre solutions de tailles et d'architectures variées seront étudiées : SAP, Business Central, Divalto et JobBoss. L'analyse de SAP et de Business Central sera réalisée de deux manières. Premièrement via la documentation officielle de chacun des deux logiciels disponible librement accessible en ligne (Windfeld Pedersen, s. d.) et ("Schema View | Production Routing | SAP Business Accelerator Hub", s. d.). Secondement, via des captures d'écrans des interfaces de configuration et de consultation. En revanche, JobBoss et Divalto étant de plus petites entités, leur documentation publique est plus restreinte. Les informations les concernant seront donc obtenues exclusivement à partir de captures d'écran des interfaces utilisateurs liées aux gammes de fabrication. De la même façon que pour le modèle de gamme commun aux MES, l'approche combinée de récupération des données pourra engendrer un certain déséquilibre dans le niveau de granularité des données recueillies, plus élevé pour SAP et Business Central. Ce déséquilibre ne posera qu'un problème limité dans le cadre de cette étude, car l'objectif est d'identifier les champs essentiels, communs ou critiques, à l'ensemble des ERP et non de modéliser en détail chaque solution. Comme pour les MES, une méthode en trois étapes sera suivie : Élimination des champs de données spécifiques au fonctionnement interne d'un ERP – Harmonisation des champs de données similaires – Évaluation de la pertinence des champs de données spécifiques.

3.2.1.3 Fusion des modèles communs MES et ERP

Les deux modèles de gammes de fabrication définis précédemment pour les MES et les ERP sont repris dans cette section. Leur mise en commun sera réalisée en harmonisant les données communes et en ajoutant celles présentes uniquement dans l'un des deux modèles, mais nécessaires, par construction, à une définition complète de la gamme. La construction de ce modèle commun de gamme de fabrication constitue une étape préalable indispensable à la suite de notre travail. Il permettra de disposer d'une structure de données commune, capable de répondre aux besoins des systèmes ERP et MES. Cependant, si ce modèle fournit une première base de structuration des champs de données nécessaires, il reste encore à le formaliser de manière rigoureuse et exploitable dans un contexte d'interopérabilité de deux logiciels. C'est pourquoi la prochaine étape de notre démarche consistera à formaliser ce modèle de gamme commun sous la forme d'une ontologie. Cette ontologie aura pour objectif de structurer cette gamme, afin de faciliter les échanges d'informations entre les différents systèmes étudiés.

3.2.2 Développement de l'ontologie

À partir du modèle de données commun aux ERP et MES établi précédemment, nous chercherons à développer une ontologie capable de formaliser de manière explicite les concepts,les propriétés et les relations associées à une gamme de fabrication. Cette ontologie occupe un rôle central dans notre approche : elle joue le rôle de structure pivot entre les deux systèmes. Elle vise à fournir une représentation sémantique commune et à favoriser l'interopérabilité en facilitant l'alignement des données échangées entre ERP et MES. Conformément aux principes du Web sémantique, l'ontologie sera formalisée dans un langage structuré permettant non seulement la représentation de la connaissance, mais aussi son exploitation automatisée par des moteurs de raisonnement. L'utilisation d'un langage normé et d'un outil reconnu dans la communauté scientifique et industrielle permettra d'assurer la pérennité, la portabilité et la reproductibilité du modèle développé.

3.2.2.1 Choix du type d'ontologie

La première étape consistera à déterminer le type d'ontologie le plus pertinent au regard des objectifs de ce mémoire. Trois grandes catégories, définies par (Guarino, 1998) et déjà citées en 2.4 seront envisagées : les ontologies de haut niveau (ou top-level), les ontologies de domaine et les ontologies d'application. L'évaluation portera sur leur capacité respective à répondre aux exigences de l'interopérabilité entre ERP et MES tout en garantissant la

réutilisabilité et la cohérence du modèle. Cette évaluation est détaillée en section 4.2.1.1 et conclue sur le choix d'une ontologie d'application. Une fois un type d'ontologie choisi, il faut déterminer un langage capable de modéliser précisément les relations sémantiques, tout en permettant une exploitation logique formelle du modèle.

3.2.2.2 Choix du langage pour l'ontologie

La deuxième étape du développement de l'ontologie porte sur le choix du langage de modélisation. Le choix du langage est une décision structurante, car il conditionne la capacité à représenter de manière précise et cohérente les concepts du domaine, ainsi qu'à exploiter pleinement les outils d'édition, de raisonnement automatique et d'interopérabilité disponibles. Un langage adapté doit permettre la structuration de la connaissance sous forme de classes, de propriétés et de relations hiérarchiques, tout en offrant des mécanismes logiques suffisamment expressifs pour capturer la complexité des processus de fabrication.

Dans ce contexte, plusieurs options seront envisagées, telles que OWL DL (Web Ontology Language Description Logic) ou RDFS (Resource Description Framework Schema) entre autres. Le choix sera guidé par une analyse des caractéristiques de chaque langage ainsi que par les usages observés dans la littérature scientifique. À l'issue de cette évaluation en section 4.2.1.2, le langage OWL DL a été retenu pour sa capacité à conjuguer expressivité et compatibilité avec les outils de raisonnement.

Finalement après avoir déterminé le type et le langage pour notre ontologie, il faudra choisir l'environnement de développement adapté.

3.2.2.3 Choix de l'outil de développement

La troisième étape du développement de l'ontologie concerne donc le choix de l'outil de développement de celle-ci. Celui-ci devra permettre la définition intuitive des entités du modèle, la visualisation des hiérarchies conceptuelles, ainsi que l'intégration de moteurs de raisonnement pour tester automatiquement la cohérence logique de l'ontologie. Pour ce faire, nous étudierons les différentes solutions existantes, le degré d'adoption académique et industrielle ainsi que leur capacité à répondre à nos besoins. La section 4.2.1.3 conclut sur l'usage du logiciel *Protégé*. Ce choix de logiciel nous permettra donc de développer le modèle commun de gamme de fabrication. Ensuite, afin de valider la pertinence du modèle ontologique développé, une phase d'évaluation expérimentale devra être menée.

3.2.3 Mise en œuvre

Afin de vérifier la qualité de l'ontologie développée, une phase d'expérimentation sera mise en œuvre dans l'avant-dernière partie. Cette étape vise à valider empiriquement la capacité du modèle à représenter fidèlement des gammes de fabrication réelles, ainsi qu'à démontrer son potentiel d'interopérabilité entre systèmes hétérogènes. Dans un premier temps, l'ontologie sera peuplée à l'aide de deux cas industriels distincts. Une première gamme de fabrication sera issue d'un logiciel MES, tandis que la seconde proviendra d'un système ERP dont les données n'ont pas été utilisées lors de la phase de construction du modèle. Cette phase permettra de tester la capacité de représentation du modèle ontologique.
Dans un second temps, nous évaluerons la pertinence de l'ontologie développée, qui se fera alors en identifiant les données issues des cas réels qui sont prises en compte ou non par notre ontologie.
Enfin, un scénario d'échange de données sera simulé afin de tester la capacité de l'ontologie à servir d'interface entre un ERP et un MES.

Ce chapitre a permis d'expliciter la problématique étudiée ainsi que la méthodologie de recherche à mettre en place pour la résoudre. Le chapitre suivant présentera alors, dans un premier temps, la construction du modèle commun de gamme en section 4.1, puis son développement sous forme d'ontologie en section 4.2.

CHAPITRE 4 CONSTRUCTION DU MODÈLE COMMUN DE GAMME DE FABRICATION

Dans ce chapitre, l'objectif est de développer le cadre d'interopérabilité entre le MES et l'ERP au niveau des gammes de fabrication. Nous décrivons, dans la suite d'abord le développement du modèle commun de gamme de fabrication en 4.1 puis le développement de l'ontologie OntoIMR issue de ce modèle 4.2. Ce qui nous permettrait d'atteindre la situation idéale illustrée en figure 4.1

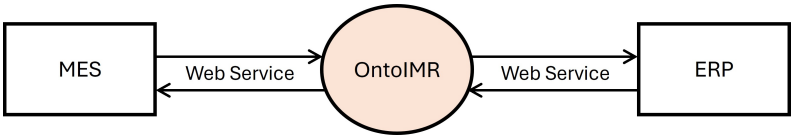

FIGURE 4.1 Schéma de communication verticale idéale

4.1 Développement du modèle commun

Pour construire ce modèle de gamme, nous appliquons donc la méthodologie illustrée à la figure 4.2 tirée de la méthodologie générale décrite au chapitre précédent.

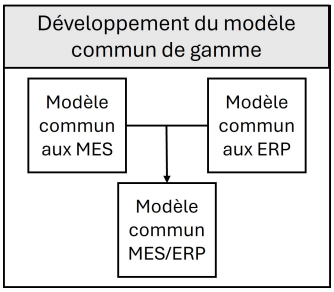

FIGURE 4.2 Schéma de la méthodologie du développement du modèle commun de gamme

L'objectif de cette section est de développer un modèle de gamme de fabrication commun aux ERP et MES, c'est-à-dire un modèle qui contient chacun des champs de données nécessaires pour la bonne description d'une gamme de fabrication dans chacun de ces types de logiciels.

Dans un premier temps, en section 4.1.1, nous nous attacherons donc à définir un modèle commun MES. Pour ce faire, nous nous appuierons en premier lieu sur la norme ISA-95, qui fournit un cadre structurant et reconnu pour la modélisation des données de production.

Ce référentiel sera complété par l'analyse de deux logiciels MES actuellement utilisés dans l'industrie manufacturière : Sepasoft et Smart Portal.

Dans un deuxième temps,en section 4.1.2 nous développerons un modèle commun aux ERP. Pour ce faire, nous nous appuierons sur quatre ERP de différentes tailles, soit SAP, BusinessCentral, Divalto et JobBOSS.

La méthode de construction des deux modèles sera similaire. La première étape consiste, pour le MES puis l'ERP, à rassembler chacun des champs de données tirés de chacun des logiciels ou normes. Dans un second temps, nous allons trier les données afin d'obtenir ce qui respecte les exigences de chacune des définitions de gamme de fabrication, conserve une capacité d'extension mais qui ne comporte pas de redondance ou de données superflues pour un modèle commun. Nous suivrons donc la démarche suivante :

1. Élimination des champs de données spécifiques au fonctionnement interne d'un logiciel : C'est-à-dire, les champs de données propres au fonctionnement interne d'un seul logiciel, sans valeur ajoutée pour la description, l'exécution, ou l'interopérabilité de la gamme, seront exclus.
2. Harmonisation des champs de données similaires : Les champs de données présentant une sémantique ou une fonction équivalente ont été regroupés ou alignés, qu'il s'agisse d'une correspondance directe ou de différences de terminologie ou de typage.
3. Évaluation des champs de données spécifiques : Les champs de données présents dans un seul des logiciels ont fait l'objet d'une évaluation quant à leur pertinence. Certains ont été retenus s'ils enrichissaient la description de la gamme, tandis que d'autres ont été exclus s'ils apparaissaient comme redondants, marginaux, ou en dehors du périmètre de cette étude.

Le schéma suivant illustre la méthode suivie pour le modèle commun MES et ERP.

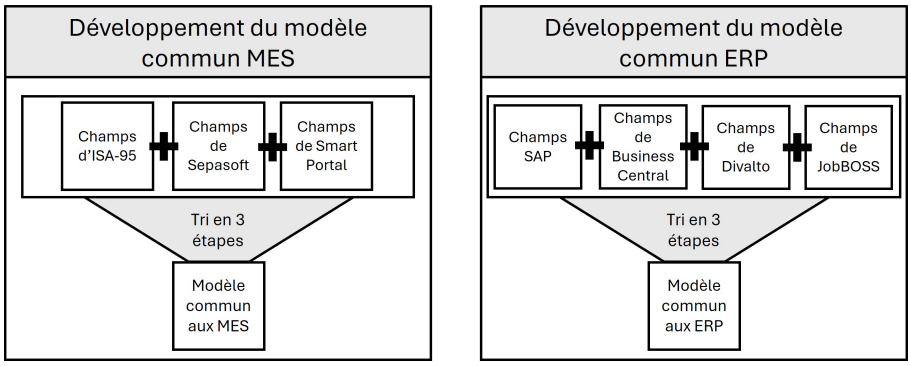

FIGURE 4.3 Schéma de la méthode du développement des modèles communs MES et ERP

Nous détaillons dans la suite ce développement pour les MES d'un côté 4.1.1 et les ERP de l'autre 4.1.2. Finalement, nous réunirons ces deux modèles afin d'obtenir un modèle de gamme de fabrication commun aux deux types de logiciels, cette étape est décrite en 4.1.3.

4.1.1 Modèle commun MES

Dans cette section, l'objectif est de construire un modèle de gamme de fabrication commun aux MES, c'est-à-dire d'établir une structure de données générique capable de couvrir à la fois les standards du secteur et les pratiques opérationnelles observées dans des systèmes MES existants, tout en restant adaptable à d'autres logiciels MES. Cette construction repose alors sur l'exploitation conjointe de la norme ISA-95 et de deux logiciels MES industriels, *Sepasoft* et *SmartPortal*. Nous décrivons dans la suite, les sources de données utilisées, le principe méthodologique employé puis les résultats obtenus.

4.1.1.1 Sources des données MES utilisées

La norme ISA-95 constitue la pierre angulaire du modèle MES développé. Cette norme internationale fournit une architecture normalisée pour l'intégration entre les systèmes d'entreprise (ERP) et les systèmes de contrôle de production (MES). Plus spécifiquement, le Product Definition Model d'ISA-95 partie 2 (*ANSI/ISA–95.00.02*, 2001) a été utilisé comme référence principale. Ce modèle propose une organisation hiérarchique des entités et des relations décrivant la fabrication, incluant la gestion des segments de processus, des ressources humaines, matérielles et d'équipement, ainsi que des paramètres de production et des spécifications de qualité.

Le logiciel *Sepasoft* propose une implémentation modulaire du MES au sein de l'environnement Ignition. La documentation technique de *Sepasoft*, en libre accès, a été exploitée pour identifier les structures de données alignées sur ISA-95 ainsi que les enrichissements propres à l'éditeur ("Track and Trace Configuration", s. d.). Il est important de noter que certains concepts prévus par ISA-95, bien présents dans la structure de *Sepasoft*, ne sont pas toujours pleinement exploités dans les configurations client standard, ce qui témoigne d'une capacité d'extension importante du logiciel.

À l'inverse, *SmartPortal* adopte une approche volontairement minimaliste. Le modèle de données repose sur un ensemble réduit de champs clés, offrant une très grande flexibilité et laissant aux entreprises la possibilité de personnaliser leur configuration selon leurs besoins spécifiques. Cette stratégie, bien que facilitant l'adaptation aux processus métiers particuliers, implique généralement un accompagnement étroit lors de l'implémentation pour compenser l'absence de standardisation exhaustive. L'analyse de SmartPortal a reposé sur : des cap-

tures d'écran fonctionnelles fournies par l'éditeur et des échanges directs avec le directeur de l'entreprise. La diversité des méthodes d'acquisition de données implique une granularité inégale des champs de données récoltés. Cependant, les données recueillies au sujet de la norme ISA-95 seront exhaustives, celles issues de la documentation officielle de Sepasoft également, et les champs de données issus de Smart Portal correspondront à un modèle de gamme minimale. Ainsi, cela ne devrait pas empêcher de construire un modèle de gamme de fabrication compatible avec la majorité des MES.

4.1.1.2 Méthodologie de mise en commun

Afin d'élaborer un modèle générique compatible avec ces 3 éléments, et potentiellement extensible à d'autres, les étapes explicitées en introduction du 4.1 ont été suivies, nous redonnons dans la suite leur titre et des exemples associés : plusieurs étapes ont été suivies :

- Élimination des champs spécifiques au fonctionnement d'un MES. Par exemple, Sepasoft enregistre deux champs, *Equipment Reference* et *Equipment Reference UUID* qui ont la même valeur sémantique, c'est-à-dire définir avec un identifiant unique un équipement mais l'un des deux est uniquement utilisé en interne par le logiciel. Ainsi, un seul est nécessaire à conserver pour conserver toute la valeur sémantique ;
- Harmonisation des champs similaires. Par exemple, un identifiant unique pouvant être un numéro ou une chaîne alphanumérique, il conserve la même valeur sémantique ;
- Évaluation des champs spécifiques. Par exemple le champs *Production visible* de Sepasoft, qui permet de définir si un élément est visible ou non depuis le plancher de production a été éliminé car il ne permet pas de définir une gamme de fabrication. De même, ISA-95 définit un grand nombre de concept liés à la maintenance des équipements. Ils ont été jugés hors du cadre de la gamme de fabrication bien que liés à la définition des équipements de production ;

4.1.1.3 Résultats de la construction du modèle

À l'issue de cette analyse, un modèle structuré autour des entités suivantes a été défini :

- Règle de Production du Produit ;
- Nomenclature de fabrication ;
- Segment de produit ;
- Paramètre du produit ;
- Dépendance du segment produit ;
- Paramètres du segment de processus ;
- Dépendance du Segment de Processus ;

- Spécification du Segment Personnel ;
- Propriété de la spécification du Segment Personnel ;
- Classe de personnel ;
- Propriétés de la classe de personnel ;
- Personne ;
- Propriété de la personne ;
- Spécification du test de compétence ;
- Attribut des résultats du test de compétence ;
- Spécification du segment d'équipement ;
- Propriétés de la spécification du segment d'équipement ;
- Classe d'équipement ;
- Propriété de l'équipement ;
- Spécification du test de capacité de l'équipement ;
- Résultat du test de capacité de l'équipement ;
- Paramètres de production ;
- Spécification du segment matériel ;
- Propriété de la spécification du segment matériel ;
- Classe de matériel ;
- Propriété de la classe de matériel ;
- Définition du matériel ;
- Propriété de la définition du matériel ;
- Spécification du test QA ;
- Résultat de test qualité ;
- Lot de matériel ;
- Propriété du lot de matériel ;
- Sous lot de matériel ;

En tout, 91 champs de données différents composent ce modèle, comme illustré dans la figure 4.4.

4.1.2 Modèle commun ERP

Parallèlement à la construction du modèle MES et avec les mêmes objectifs en tête, nous nous attachons dans cette section à la construction d'un modèle commun de gamme de fabrication commun aux ERP. Pour ce faire, nous allons utiliser les données de 4 logiciels ERP de tailles et d'architectures différentes : SAP, Business Central, Divalto et JobBoss.

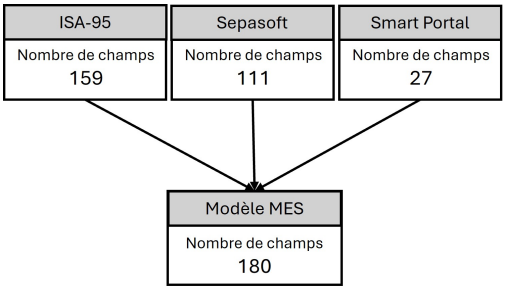

FIGURE 4.4 Nombre de champs de données en fonction des sources MES

4.1.2.1 Sources des données ERP utilisées

SAP est l'un des ERP les plus implantés à l'échelle mondiale, notamment dans les grandes entreprises industrielles. Sa structure de données est extrêmement riche, proposant des modèles de gamme de fabrication très détaillés, intégrant la gestion des opérations, des composants, des ressources humaines et matérielles, ainsi que des spécifications de qualité. La documentation officielle ("Schema View | Production Routing | SAP Business Accelerator Hub", s. d.) et des échanges avec des utilisateurs expérimentés (accompagnés de captures d'écran) ont permis d'accéder à une compréhension fine de son modèle de données. *Microsoft Dynamics 365 Business Central* est un ERP plus léger que SAP, conçu pour les PME et les entreprises de taille intermédiaire. Il propose une structure de gamme moins complexe, mais suffisamment modulable pour couvrir la majorité des besoins de fabrication. La documentation en ligne officielle (WINDFELD PEDERSEN, s. d.) ainsi que des retours d'expérience d'utilisateurs avancés (accompagnés également de captures d'écran) ont été exploités pour analyser sa manière d'organiser les données de production. *Divalto Infinity* est un ERP français principalement destiné aux PME industrielles. Sa documentation n'étant pas librement accessible, l'analyse s'est appuyée sur des captures d'écran des interfaces utilisateurs liées aux gammes de fabrication tirées d'exemples d'une firme de consultation spécialisée dans son intégration. Divalto propose une structuration relativement simple de la gamme mais il offre une capacité d'adaptation via des personnalisations métiers. *JobBoss* est un ERP nord-américain spécifiquement conçu pour les petites structures manufacturières (ateliers de production, fabricants sur commande). Le logiciel privilégie une approche pragmatique et opérationnelle, avec des modèles de gamme épurés et très orientés "terrain". Comme pour Divalto, les informations ont été obtenues par l'analyse directe d'exemples d'utilisation issues d'une firme intégratrice de cette solution via des captures d'écrans des interfaces utilisateur. Cette diversité de solutions a permis de croiser des modèles très riches (SAP, Business Central) avec des approches plus compactes et adaptées à des contextes industriels plus souples (Divalto, JobBoss).

Il est important de noter que, comme pour la construction du modèle MES, la disponibilité des sources a induit une granularité variable des données collectées :

— SAP et Business Central ont permis d'accéder à des descriptions de gamme très détaillées, enrichies de nombreuses métadonnées ;
— tandis que Divalto et JobBoss ont fourni une vision plus synthétique, correspondant aux besoins essentiels d'atelier et à des besoins très accès terrain ;

Cette hétérogénéité a été prise en compte dans la méthodologie d'élaboration du modèle commun afin de garantir sa capacité d'adaptation à différents niveaux de complexité industrielle.

4.1.2.2 Méthodologie de mise en commun

Afin d'élaborer un modèle ERP générique, capable d'intégrer les spécificités des quatre logiciels étudiés tout en restant adaptable à d'autres environnements, la méthode suivie pour les MES a été répétée, sa description détaillée est visible en 4.1.1

— **Élimination des champs spécifiques** au fonctionnement interne :Par exemple, dans SAP, le champ *LongTextLanguageCode*, qui précise uniquement la langue du texte long associé à une gamme, a été considéré comme hors périmètre et écarté ;
— **Harmonisation des champs similaires** : Par exemple, les format d'identifiant ou les différence de types de donnée sont été alignés et regroupés sous des définitions communes ;
— **Évaluation des champs spécifiques** : Par exemple, parmi les nombreux champs liés à la gestion de la qualité dans SAP, seuls les champs définissant les caractéristiques d'inspection ont été conservés. Les champs décrivant en détail les méthodes d'inspection ont été considérés comme des extensions optionnelles, non incluses dans ce modèle de base ;

Cette méthodologie progressive a permis de tenir compte des différences de granularité observées entre les 4 ERP. Ainsi, le modèle construit vise à assurer une couverture fonctionnelle étendue tout en conservant la flexibilité nécessaire pour s'adapter à des environnements industriels variés. Le modèle reste par construction centré sur la description de la gamme, sans inclure d'éléments de planification (ordonnancement, contraintes de ressources dynamiques, etc.),il reste cependant extensible en cas de besoins futurs.

4.1.2.3 Résultats de la construction du modèle

À l'issue de l'analyse des quatre ERP étudiés et de l'application de la méthodologie décrite précédemment, un modèle ERP générique a été défini, structuré autour d'entités fondamen-

tales nécessaires à la description d'une gamme de fabrication. Ce modèle se compose des entités suivantes :

— Entête de gamme de fabrication ;
— Séquence ;
— Opération ;
— Affectation du Produit ;
— Affectation des composants d'opération ;
— Affectation diverse PRT ;
— Affectation de document PRT (document technique, plans, manuels techniques) ;
— Affectation PRT matérielle (matériau utilisé temporairement, ou consommable) ;
— Affectation des équipement PRT (équipement ou machine) ;
— Caractéristique d'inspection ;

Ce modèle regroupe 134 champs de données différents, répartis entre ces entités, offrant ainsi une structure détaillée mais modulaire. La répartition du nombre de champs issus de chaque ERP est présentée à la figure 4.5 .

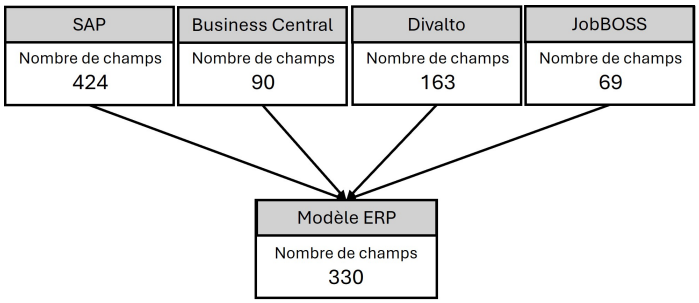

FIGURE 4.5 Nombre de champs de données en fonction des sources ERP

4.1.3 Modèle Commun

À l'issue de la construction des modèles spécifiques aux MES et aux ERP, une étape de consolidation a été entreprise pour élaborer un modèle commun de gamme de fabrication, visant à servir de pivot entre les deux environnements. Cette mise en commun repose sur une approche structurée, visant à maximiser la compatibilité tout en assurant l'exhaustivité nécessaire à la description d'une gamme.

4.1.3.1 Principes de construction

Les deux modèles précédemment définis ont été comparés de manière systématique. La construction du modèle commun a suivi deux principes :

— Regrouper les données communes entre ERP et MES, afin de constituer un socle d'informations partagées ;
— Intégrer les données spécifiques présentes uniquement dans l'un des deux modèles, dès lors qu'elles s'avéraient nécessaires pour assurer une définition complète, précise et exploitable de la gamme de fabrication ;

Cette approche permet de garantir que le modèle commun conserve l'intégralité des éléments critiques nécessaires à la gestion, à l'exécution et au pilotage des processus de fabrication, quels que soient les systèmes en interaction.

4.1.3.2 Cas paticulier : Harmonisation de la gestion des séquences

Un point particulier a nécessité une attention approfondie : la méthode de définition de l'enchaînement des opérations. Dans les modèles ERP, la séquence est représentée comme une entité dédiée, associant chaque opération à un ordre d'exécution explicite (par exemple via des numéros de séquence ou des liens directs entre étapes). Dans les modèles MES, notamment inspirés d'ISA-95, l'ordre des opérations est plutôt déterminé par des dépendances entre opérations, exprimées sous forme de relations de précédence. Cette différence n'affecte pas tant les données disponibles que la manière conceptuelle dont l'enchaînement est structuré.

Afin de concilier ces deux logiques et de conserver une compatibilité maximale avec les pratiques ERP et MES, il a été décidé de maintenir une entité *Sequence* propre, permettant d'associer explicitement un ordre aux opérations et de préserver une entité *Process Segment Dependency*, décrivant les relations potentielles de précédence entre opérations. Ainsi, le modèle commun est capable d'exprimer directement un ordre séquentiel simple (comme attendu par de nombreux ERP) tout en étant suffisamment flexible pour décrire des structures plus complexes de dépendances (comme celles gérées dans les MES). Ce double mécanisme assure une capacité d'adaptation accrue en fonction du système cible et des besoins métiers spécifiques.

4.1.3.3 Résultats

Le modèle commun ERP/MES regroupe l'ensemble des champs jugés essentiels à la description d'une gamme interopérable. Il se compose des entités structurantes suivantes (*l'italique* indique que le nom est issu d'ISA-95 partie 2 (*ANSI/ISA–95.00.02*, 2001) :

— *Product production rule* ;
— *Process segment* ;
— Sequence ;
— Process segment Dependency ;

— Process Segment Product Allocation ;
— Process Segment Component Allocation ;
— Miscellaneous PRT Assignment ;
— Document PRT Assignement ;
— Material PRT Assignment ;
— Equipment PRT Assignment ;
— Inspection Specification ;
— Personal Specification ;

Ce modèle regroupe 193 champs de données différents, répartis grâce à ces entités.
Le nombre de champs issus de chacun des modèles de départ (ERP et MES) est présenté à la 4.6. On peut remarquer que le nombre de champs ajoutés par chaque logiciel est plus faible que ceux du premier logiciel, ce qui montre bien la structure commune à la base de chaque gamme de fabrication dans les logiciels. Si l'on regarde plus en détail les MES, on peut voir que l'on a enrichi le modèle ISA-95 de gammes de fabrication considéré de 40 champs pour le modèle MES, soit de 28,5% de ses champs. Par ailleurs, 22,2% des champs du modèle MES ont été ajoutés pour compléter le modèle tiré de l'ERP et donc 78,8% de ses champs étaient en commun avec le modèle ERP.

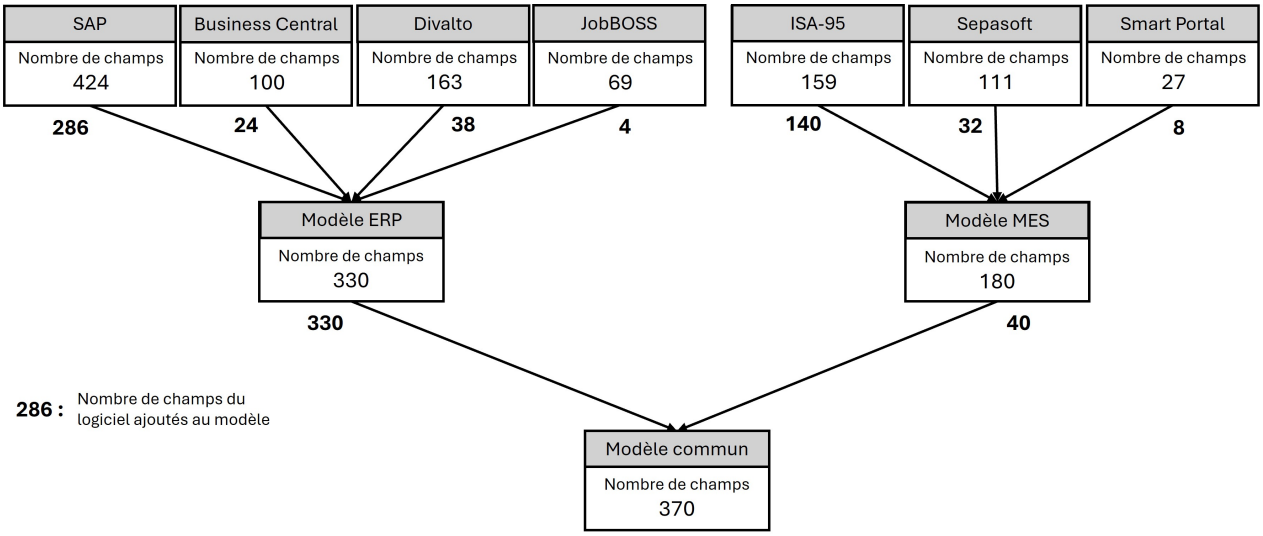

FIGURE 4.6 Nombre de champs de données du modèle commun

Ce modèle commun constituera la base pour la construction de l'ontologie de gamme de fabrication, permettant d'assurer une interopérabilité sémantique robuste entre ERP et MES.La section suivante décrit l'utilisation de ce modèle pour la réalisation de l'ontologie.

4.2 Construction de l'ontologie

Le modèle commun de gamme de fabrication nous permet désormais de construire l'ontologie pour représenter formellement les concepts clés de la gamme et des relations qui les unissent. Le recours à une ontologie permet d'aller au-delà d'une simple correspondance syntaxique des données : il s'agit ici de capter et de structurer la signification des éléments échangés afin de renforcer la compréhension mutuelle entre applications hétérogènes. Ce modèle se distingue notamment par sa flexibilité pour faire évoluer la structure des données, sa capacité à centraliser l'information dans un cadre conceptuel unifié et à favoriser l'interopérabilité. Pour cela, l'ontologie a été conçue selon les principes du Web sémantique, en utilisant un langage adapté à la représentation des connaissances formelles et à leur traitement automatisé par des outils de raisonnement. Nous allons voir dans un premier temps les choix des caractéristiques de l'ontologie (type, langage et outil). Dans un second temps, nous détaillerons la modélisation des concepts clés, de leurs champs associés et des relations qui les lient. Nous suivrons donc la méthode explicitée en 3.2.2, illustrée à la figure 4.7

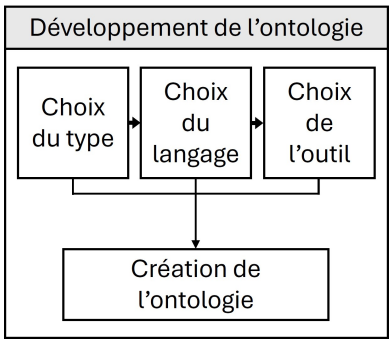

FIGURE 4.7 Schéma de la méthodologie du développement de l'ontologie

4.2.1 Choix des caractéristiques de l'ontologie

La conception d'une ontologie ne repose pas uniquement sur le contenu qu'elle représente, mais également sur les choix techniques et conceptuels qui en déterminent la portée, l'utilisabilité et la robustesse. Trois décisions structurantes ont été prises avant d'entamer la phase de modélisation proprement dite : le type d'ontologie à adopter, le langage de formalisation à privilégier, et l'outil utilisé pour sa conception.

4.2.1.1 Choix du type d'ontologie

Dans la littérature et notamment (Guarino, 1998) plusieurs catégories d'ontologies sont distinguées selon leur niveau d'abstraction et leur finalité. On retrouve principalement :

- Les ontologies de haut niveau (ou upper ontologies) qui visent à formaliser des concepts généraux et indépendants de tout contexte spécifique, comme le temps, l'espace ou l'événement ;
- Les ontologies de domaine, qui formalisent les connaissances propres à un champ d'activité donné — ici, le secteur manufacturier ;
- Les ontologies d'application, conçues pour modéliser les entités et interactions propres à un cas d'usage particulier, souvent en lien avec une solution logicielle ou un besoin opérationnel ;

Dans le cadre de ce travail, l'objectif est de modéliser avec précision les données et interactions nécessaires au couplage MES–ERP, en se concentrant sur la structure des gammes de fabrication. Ce besoin implique une forte contextualisation, à la fois technique et métier. Le choix s'est donc porté sur une **ontologie d'application**, qui offre le niveau de détail requis pour représenter les concepts spécifiques à notre cas d'étude, tout en demeurant opérationnelle et orientée vers l'intégration. Comme le souligne (Guarino, 1998), ce type d'ontologie repose sur une vision pragmatique de la modélisation, centrée sur l'adéquation avec un usage concret, ce qui correspond pleinement aux exigences de notre approche.

Le type d'ontologie étant désormais défini, la section suivante s'intéresse au choix du langage, en tenant compte des capacités d'expression sémantique et de raisonnement nécessaires à l'exploitation formelle du modèle.

4.2.1.2 Choix du langage

Le choix du langage ontologique a été orienté par plusieurs critères : le niveau d'expressivité sémantique, la compatibilité avec les moteurs de raisonnement, le degré de normalisation, ainsi que l'adoption dans les environnements industriels. Plusieurs langages ont été étudiés en regard de ces exigences.

Le langage *RDFS* ("RDF Schema 1.1", 2014), bien qu'élément de base du Web sémantique, se limite à des hiérarchies simples de classes et de propriétés. Il ne permet pas d'exprimer des contraintes avancées telles que les restrictions de cardinalité, les équivalences de classes ou les relations de disjonction, le rendant peu adapté à la représentation de processus industriels complexes.

le langage *OWL DL* "OWL 2 Web Ontology Language Primer (Second Edition)", 2012. Il constitue un compromis robuste entre expressivité et décidabilité, en permettant la définition

de classes, de relations, de contraintes logiques complexes et d'inférences automatisées, tout en restant conforme aux standards du Web sémantique du W3C. Son adoption dans de nombreux outils, dont Protégé et les raisonneurs comme HermiT ou Pellet, en fait une solution mature et bien documentée pour les besoins industriels.
Le langage *OWL Full* "OWL 2 Web Ontology Language Primer (Second Edition)", 2012 offre une expressivité maximale en combinant les possibilités de RDF et OWL sans restrictions syntaxiques. Toutefois, cette flexibilité se fait au détriment de la décidabilité : les moteurs de raisonnement ne peuvent garantir l'achèvement ni la cohérence des inférences, ce qui pose problème dans un contexte d'intégration opérationnelle.
Des langages plus syntaxiques tels que RDF/XML ou XSD "RDF 1.1 XML Syntax", 2014 permettent une structuration formelle des données, mais n'intègrent pas de sémantique ni de capacités de raisonnement. Leur usage se limite à la validation de structure, et non à la modélisation de connaissances.
Le tableau 4.1 résume les principaux critères de comparaison des langages envisagés. Finalement, le choix du langage se porte sur OWL DL qui correspond le mieux à nos attentes. Une fois ce langage choisi, déterminons l'outil ou le logiciel de développement de l'ontologie.

4.2.1.3 Choix de l'outil

Le choix de l'environnement de modélisation repose sur plusieurs critères : compatibilité avec le langage OWL DL, support de moteurs de raisonnement, accessibilité, interopérabilité avec d'autres outils du Web sémantique, ainsi qu'une large adoption dans la communauté scientifique. L'outil devait aussi permettre une modélisation intuitive, tout en garantissant la rigueur formelle nécessaire à la définition d'axiomes logiques. Au regard de ces exigences, *Protégé* a été retenu comme environnement de développement. Développé et maintenu par l'Université de Stanford, c'est un logiciel libre largement adopté dans le milieu académique et industriel. Il permet la création intuitive d'ontologies via une interface graphique, tout en offrant la possibilité d'exporter les modèles au format OWL. *Protégé* se distingue également par l'intégration native de moteurs de raisonnement tels que HermiT, Pellet ou FaCT++, permettant de vérifier automatiquement la cohérence de l'ontologie et d'inférer des connaissances implicites.Enfin, sa compatibilité avec d'autres outils du Web sémantique (éditeurs RDF, plateformes de requêtes SPARQL, outils d'alignement ou d'évaluation) en fait un outil de référence pour la modélisation sémantique.
Ce choix d'outil marque ainsi la dernière étape de la définition des caractéristiques structurelles de notre ontologie. Dans la suite, nous nous appuierons sur cette base pour modéliser les classes et les relations représentant les éléments de la gamme de fabrication commune définie en 4.1.3.

Tableau 4.1 Tableau comparatif des langages ontologiques

Langage	Expressivité	Raisonnement	W3C	Avantages	Limites
OWL DL	Élevée	Oui	Oui	Bon compromis entre expressivité et décidabilité	Courbe d'apprentissage technique
OWL Full	Très élevée	Non garanti	Oui	Très expressif, peu de contraintes syntaxiques	Raisonnement non décidable
RDFS	Faible	Limité	Oui	Simple, largement adopté	Trop limité pour des structures complexes
RDF/XML	Faible	Non	Oui	Syntaxe simple pour triplets RDF	Pas adapté à la modélisation logique
XSD	Faible	Non	Non	Structuration syntaxique XML claire	Pas de capacités sémantiques

4.2.2 Développement ontologique

Afin de construire notre ontologie, nous allons donc utiliser le logiciel *Protégé*. Ce logiciel permet d'implémenter une ontologie via ses constituants : les concepts appelés *Classes*, les relations appelées *Object properties*, les propriétés de données appelées *Data Properties* et les instances appelées *Individuals*. Un guide de création de première ontologie sur Protégé explique leur fonctionnement et leur implémentation (Noy et McGuinness, 2001).

4.2.2.1 Définition des concepts

Les *Classes* représentent les concepts fondamentaux de l'ontologie, c'est-à-dire ici les entités manipulées dans le cadre d'une gamme de fabrication. Chaque *Classe* est associée à un certain nombre de *Data properties* qui permettent de qualifier ses instances à l'aide de valeurs littérales, qui peuvent être des nombres, des chaînes de caractères, des dates, etc. Le tableau en annexe B présente l'intégralité des *Classes* de notre modèle ainsi que leurs *Data properties* explicitement associées.

Afin d'illustrer la construction de notre modèle, nous suivrons dans cette section et la suivante l'exemple de la classe *Resource* et de ses classes associées. Le tableau 4.2 présente 10 des 62 *Classes* de notre modèle, leur définition, les *Data Properties* qui leur sont associées et leur éventuelle hiérarchie. Les termes en italique dans le tableau sont directement issus de la norme ISA-95.

Tableau 4.2 Decription d'une partie des *Classes* et de leur *Data properties* de l'ontologie

Classes	**Définition**	**Champs associés**	**Sous-classe de**
Resource	Ensemble des moyens nécessaires à l'exécution d'une opération.	*Name*, Description, *Comment*, ChangedDateTime, ChangeNumber, CreationDateTime	
Personal Specification	Ensemble des caractéristiques associées à une ressource humaine.	Person_ID, WageGroup, WageType	Resource
PRT	Ensemble des ressources non humaine.	ItemNo	Resource
PRT_Material	Matériau requis au cours d'une opération, non inclu dans la BOM.	ID	PRT
PRT_Equipment	Équipement utilisé pendant une opération.	ID	PRT
PRT_Document	Document de référence utilisé lors d'une opération.	ID	PRT
DocumentPart	Élément constitutif d'un document de production, permettant de référencer de manière plus précise ses sections.	ID	PRT_Document
PRT_Miscellaneous	Ressource temporaire ne relevant d'aucune catégorie prédéfinie.	ID	PRT
ResourceType	Catégorie de classification des ressources facilitant leur classification et leur gestion.	ResourceType, Name, Description, Comment, ChangedDateTime, ChangeNumber, CreationDateTime	
Version	Instance décrivant une version donnée d'un objet du modèle incluant sa période de validité.	No, ValidityEndDate, ValidityStartDate	

Les figures 4.8 et 4.9 viennent compléter ce tableau en présentant respectivement l'organisation de ces classes et celle de leurs Data Properties sur *Protégé*. Sur la figure 4.8 on observe que la classe *Resource* possède une hiérarchie relativement riche, avec une profondeur de 4. En ce qui concerne les *DataProperties* un choix a été fait : certaines sont génériques et peuvent être utilisées par n'importe quelle classe sans implication sémantique forte, comme *Description*, *Name*, *Comment* etc. D'autres, en revanche, ont un domaine spécifique qui contraint leur usage et structure davantage l'ontologie, comme Person_ID visible sur la figure 4.9.

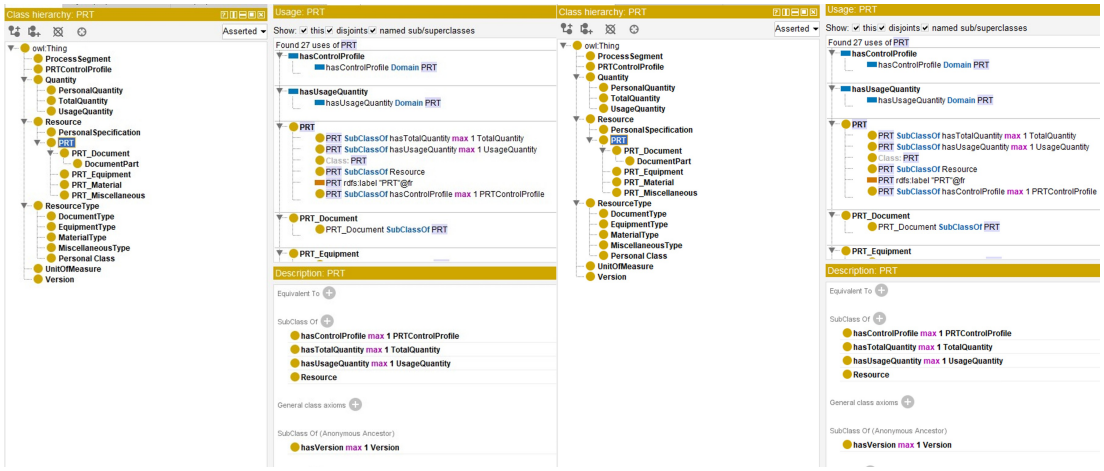

FIGURE 4.8 Interface du logiciel *Protégé* sur la vue des *Classes* pour les ressources

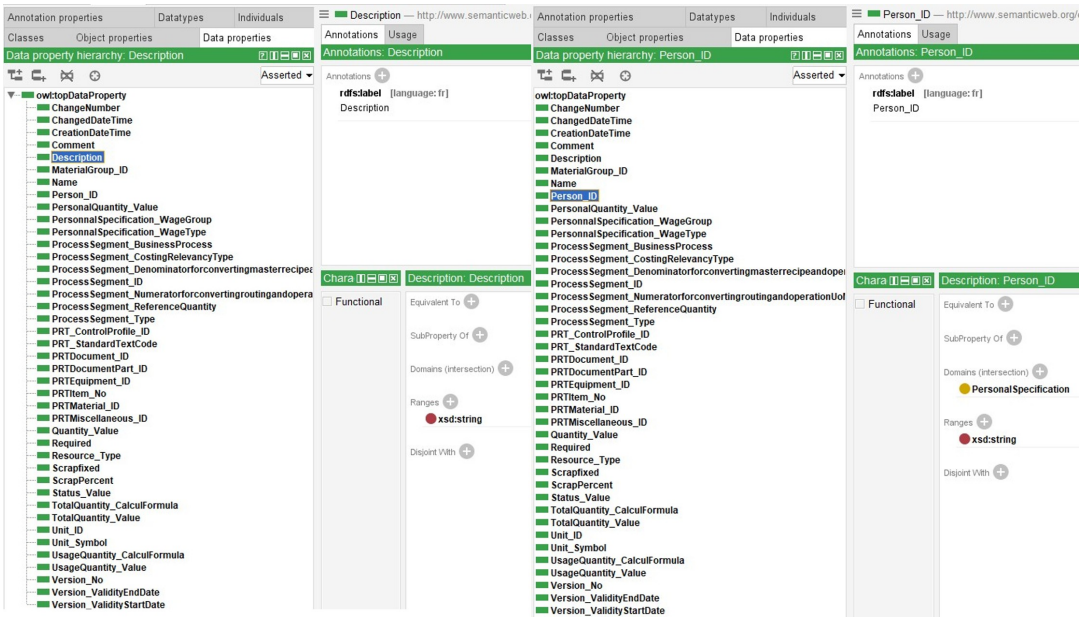

FIGURE 4.9 Interface du logiciel *Protégé* sur la vue des *DataProperty* des *Resources*

L'objectif de cette structuration est de représenter le plus fidèlement possible les attributs

du modèle de gamme commun défini en 4.1.3 tout en assurant à la fois l'interopérabilité sémantique et une grande flexibilité pour adapter le modèle à différents logiciels.

4.2.2.2 Modélisation des relations

Les *Object Properties* permettent d'exprimer les relations entre les différentes classes. Elles jouent un rôle clé dans la structuration de l'ontologie, en spécifiant comment les entités interagissent ou sont liées entre elles.

Chaque propriété d'objet peut être définie avec un domaine de définition (*Domain*) et une portée (*Range*), ce qui garantit la validité logique des liens dans l'ontologie. Nous allons expliciter cela en poursuivant l'exemple de la classe *Resource* : sur la figure 4.10 la relation *IsusedInProcessSegment* a pour domaine la classe *Resource* et pour portée la classe *ProcessSegment*, ce qui permet d'associer à une opération une ressource nécessaire à sa réalisation. Cependant, il est important de bien comprendre le rôle de ces déclarations. Dire que la propriété *IsusedInProcessSegment* a pour domaine *Resource* et pour portée (range) *ProcessSegment* ne signifie pas que seuls des individus de la classe *Resource* peuvent apparaître comme sujets, ni que seuls ceux de *ProcessSegment* peuvent être objets. Cela veut simplement dire que si un individu est lié à un autre par cette propriété, alors le premier est inféré comme appartenant à la classe *Resource*, et le second à la classe *ProcessSegment*. Cette règle est exploitée par les moteurs de raisonnement pour inférer automatiquement les classes des individus. La définition des domaines et portée n'est pas obligatoire. Certaines relations n'en possèdent pas, notamment lorsque plusieurs classes peuvent être utilisées comme domaine ou portée. Par exemple, la relation *Version* est définie avec une portée *Version* mais sans domaine, car de nombreuses classes peuvent l'utiliser. Pour renforcer la lisibilité sémantique, on peut également définir des relations à l'aide de restrictions sur les sous-classes. Par exemple, pour exprimer que tous les *ProcessSegment* utilisent au moins une *Resource*, on peut déclarer que *ProcessSegment* est une sous-classe de *usesResource some Resource*. Cela signifie que tout individu de la classe *ProcessSegment* doit être relié à au moins un individu de la classe *Resource* par la propriété *usesResource*. Cette approche est utile pour que les moteurs de raisonnement puissent exploiter pleinement la logique de l'ontologie. Il est également possible de caractériser les relations entre elles, par exemple en spécifiant que *isUsedInProcessSegment* est l'inverse de *usesResource*. Cela signifie que si une ressource est utilisée dans un *ProcessSegement*, alors ce segment utilise cette ressource. Sans cette déclaration, l'implication ne serait pas automatique. Enfin, les cardinalités permettent de restreindre le nombre de relations qu'un individu peut entretenir via une propriété. On distingue plusieurs formes : *some* (au moins une), *min* (au moins n), *max* (au plus n) et *exactly* (exactement n). L'absence

de contrainte rend une *object property*. Ces contraintes renforcent la précision du modèle et permettent aux raisonneurs de vérifier sa cohérence mais elles le rendent aussi plus rigide. Dans notre cas, le modèle vise à rester très flexible pour s'adapter aux structurations des gammes de fabrication de plusieurs logiciels. Nous avons donc volontairement défini peu de cardinalités, notamment en évitant celles qui imposeraient la présence obligatoire d'une donnée (comme *min ou some*). Le tableau 4.3 présente 4 des 44 *Object Properties* de notre modèle, liées à la classe *Resource*.

TABLEAU 4.3 Descriptif des Object Properties liés à *Resource*

Object Properties	**Domain**	**Range**	**Cardinalité**
has Version		Version	max 1
hasResourceType	Resource	Resource Type	None
usesResource	Process Segment	Resource	None
isUsedIn ProcessSegment	Resource	Process Segment	None

Nous pouvons alors visualiser la représentation de la classe *Resource* et de ses différentes relations à l'aide d'OntoGraf, comme illustrées à la figure 4.10 et figure 4.11.

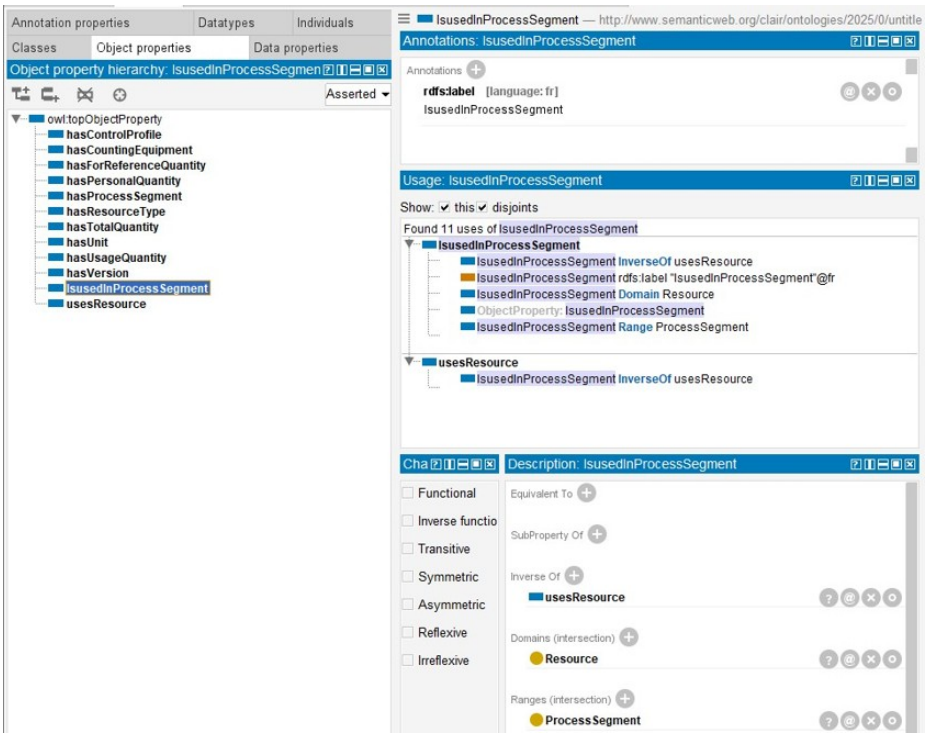

FIGURE 4.10 Interface du logiciel *Protégé* sur la vue des *Object Properties* des *Resource*

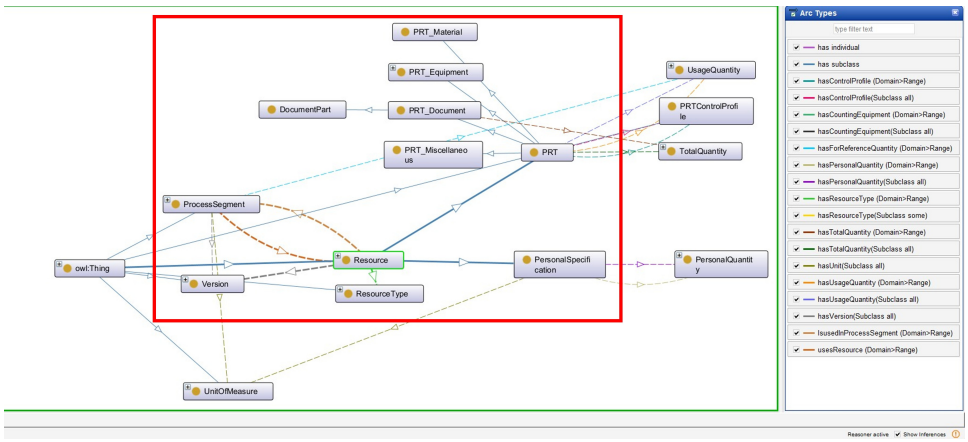

FIGURE 4.11 Visualisation de l'Ontograf centré sur les *Resource*

La modélisation de l'ontologie dans Protégé a permis de structurer les concepts clés liés à la gamme de fabrication sous forme de classes, de propriétés et de relations, tout en respectant les principes de cohérence logique et sémantique. Les règles de domaine, de portée et de cardinalité ont été utilisées afin de garantir une représentation fidèle des interactions entre les entités du modèle mais également de conserver une grande flexibilité. Cette modélisation constitue ainsi une base solide et exploitable pour assurer l'interopérabilité entre systèmes ERP et MES au niveau des gammes de fabrication.

4.3 Conclusion

La construction du modèle commun a permis d'unifier les structures issues des systèmes ERP et MES, en identifiant les entités et les relations essentielles à la description d'une gamme de fabrication. Ce modèle a ensuite été formalisé sous la forme d'une ontologie développée dans Protégé accessible au lien suivant (https ://github.com/ClaireJodeau/OntoIMR.git), afin d'en assurer une représentation explicite, normalisée et directement exploitable par des outils de raisonnement automatique. Les principales caractéristiques de cette ontologie sont présentées dans le tableau 4.4, tandis qu'une vue d'ensemble est illustrée à la figure 4.15 et les modélisations complètes des *Class*, *Object Properties* et *Data Properties*, respectivement aux figures 4.12, 4.13 et 4.14.

La section suivante sera consacrée à la validation de cette ontologie, tant du point de vue structurel que sémantique, dans le but d'en évaluer la robustesse, la cohérence et la capacité à modéliser fidèlement des cas industriels réels.

Tableau 4.4 Descriptif de l'ontologie

Caractéristique	**Valeur**
Nombre de *Classes*	62
Nombre de classes racines	31
Profondeur hiérarchique maximale	4
Nombre d' *Object Properties*	44
Nombre de *Data Properties*	93

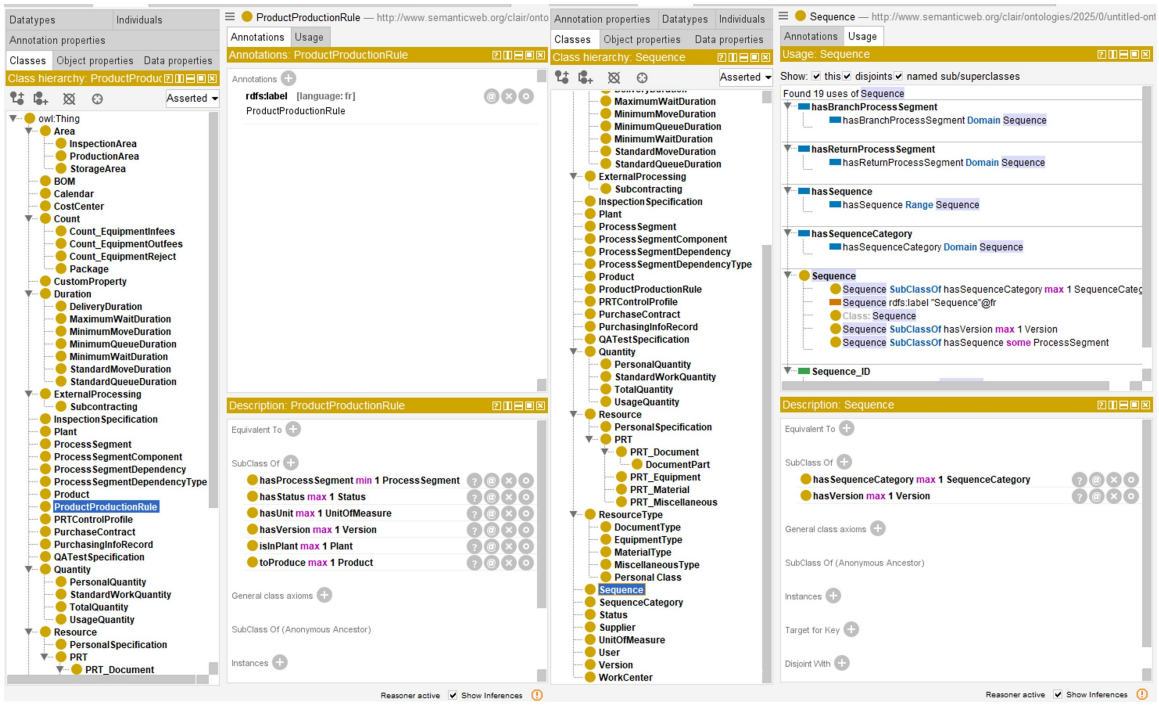

Figure 4.12 Interfaces de Protégé sur les Classes

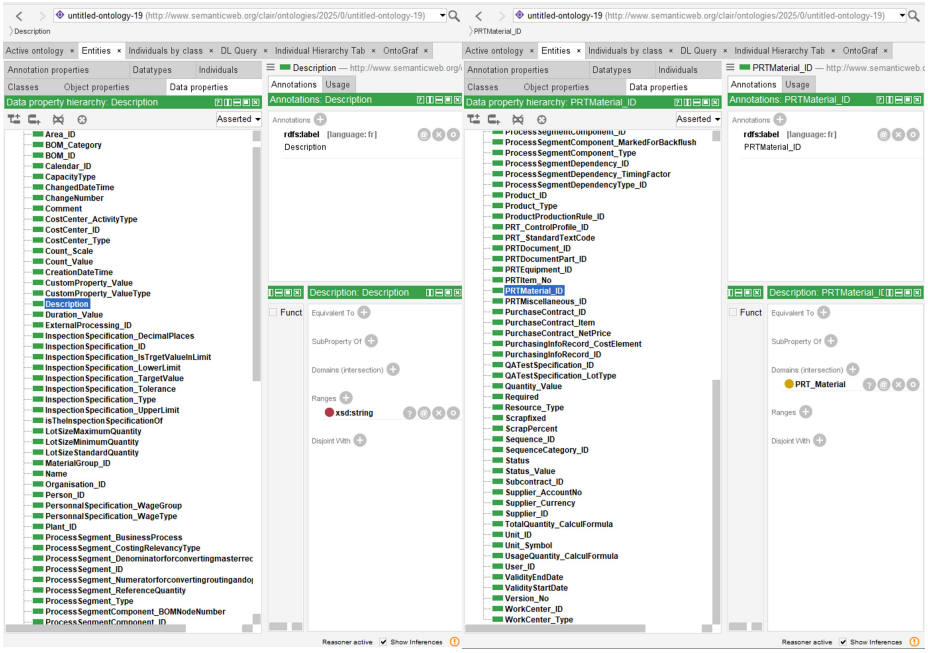

FIGURE 4.13 Interfaces de Protégé sur les *Data Properties*

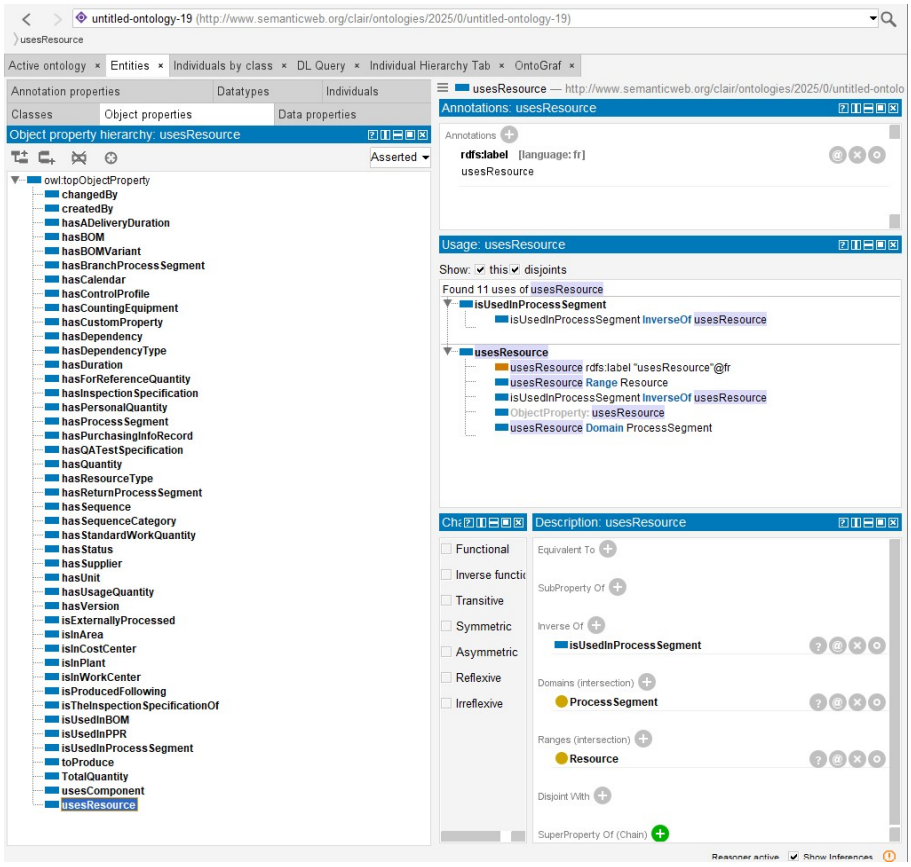

FIGURE 4.14 Interfaces de Protégé sur les *Object Properties*

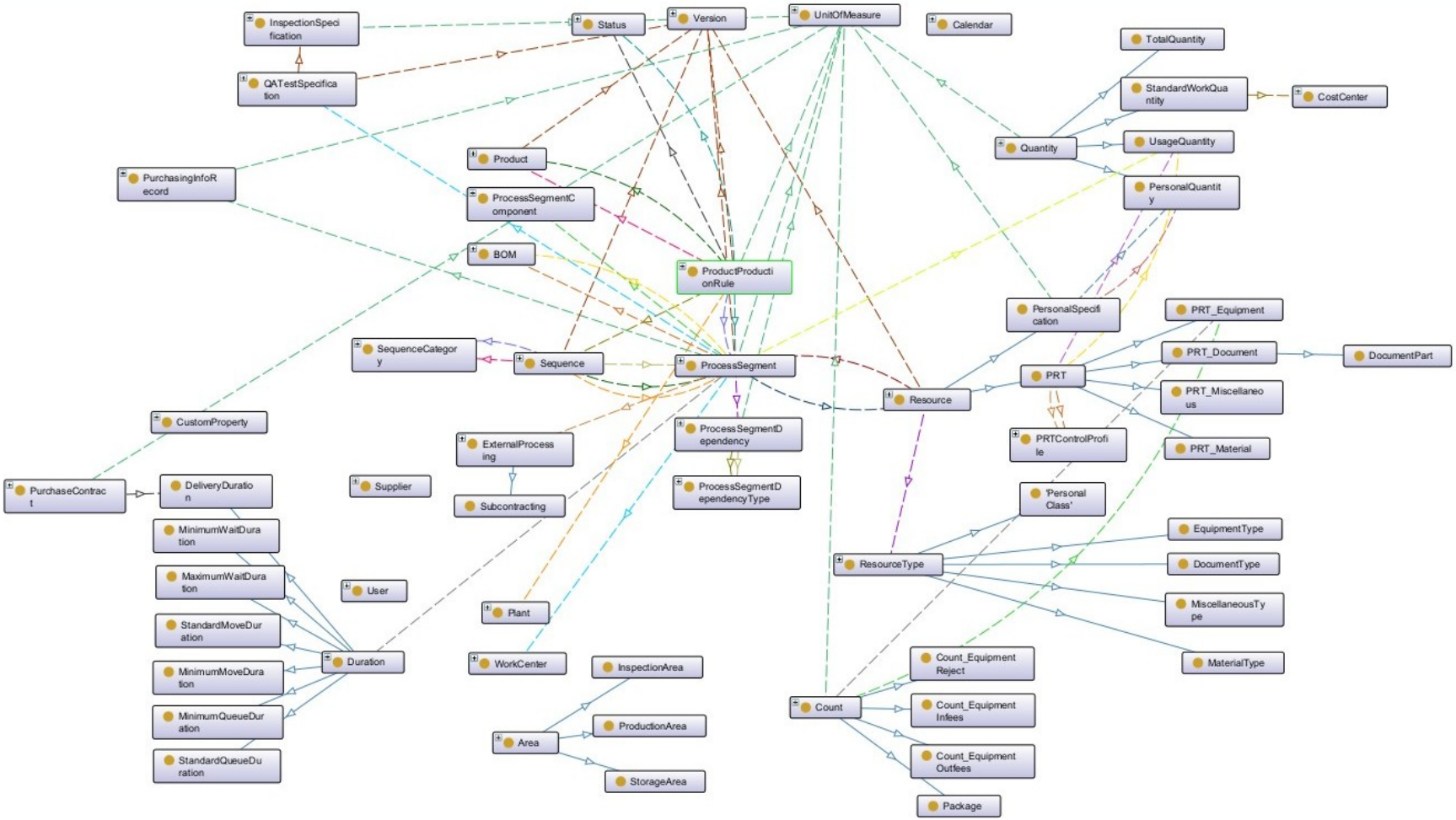

FIGURE 4.15 Vue globale (OntoGraf) de l'ontologie, représentation simplifiée avec concepts et liens

CHAPITRE 5 VALIDATION

Les chapitres précédents ont permis de présenter la méthodologie suivie dans le cadre de cette recherche, ainsi que le processus de développement de l'ontologie proposée. Celle-ci vise à faciliter le couplage entre les systèmes MES et ERP, en offrant un modèle commun pour la représentation des gammes de fabrication et un cadre d'interopérabilité le supportant. Le modèle idéal d'utilisation a été illustré à la figure 4.1. Une fois cette phase de modélisation achevée, il est indispensable d'engager une démarche de validation afin de s'assurer que l'ontologie répond aux exigences de cohérence formelle et de pertinence métier. Cette validation a pour objectif de détecter d'éventuelles erreurs de modélisation susceptibles d'entraver son utilisation dans un environnement applicatif, d'évaluer sa capacité à représenter fidèlement les concepts du domaine industriel et de proposer une preuve de concept de sa capacité à être un outil pivot entre MES et ERP.

5.1 Méthodologie

La validation repose sur une approche en deux volets complémentaires, permettant de couvrir à la fois la dimension technique et la dimension fonctionnelle du modèle développé :

- Une validation structurelle en 5.2, centrée sur la cohérence logique, la qualité syntaxique et la conformité du modèle aux standards du langage OWL ;
- Une validation sémantique en 5.3, axée sur sa capacité à représenter avec précision des situations concrètes issues du couplage MES–ERP. Cette approche s'inspire notamment des recommandations méthodologiques formulées par Obrst et al. (2006) en matière d'évaluation ontologique ;
- Une preuve de concept en 5.4 montrant la capacité d'OntoIMR à recevoir et renvoyer les informations issus et de l'ERP et du MES ;

5.2 Validation structurelle

Le premier axe de validation porte sur les aspects structurels de l'ontologie. L'objectif est de s'assurer que le modèle est formellement cohérent, qu'il respecte les règles syntaxiques du langage OWL, et qu'il ne présente aucune anomalie susceptible de compromettre son exploitation dans un contexte applicatif, quel qu'il soit. Cette étape permet également de vérifier que le modèle suit les bonnes pratiques en matière de modélisation ontologique, facilitant ainsi sa maintenance, sa réutilisation et sa possible extension. Plus précisément, la

validation structurelle vise à garantir :

— de l'absence d'incohérences logiques entre les axiomes ;
— de la bonne structuration hiérarchique des classes ;
— de la définition correcte des propriétés (domaines, ranges, cardinalités) ;
— de l'absence de redondances ou de pièges de modélisation ;

5.2.1 Outils utilisés pour la validation structurelle

Pour mener à bien cette analyse, plusieurs outils et descripteurs logiques ont été mobilisés. Ces outils s'appuient sur les règles formelles définies dans Protégé et sur les standards OWL pour détecter automatiquement les incohérences ou irrégularités éventuelles :

— **HermiT** raisonneur basé sur les logiques de description, intégré par défaut dans Protégé. Il permet de détecter les incohérences, de classifier automatiquement les concepts, et d'inférer des sous-classes implicites ;
— **Pellet** raisonneur complet prenant en charge OWL DL, offrant également des fonctionnalités d'explication des incohérences et de vérification de la satisfabilité des classes ;
— **OOPS !** (Ontology Pitfall Scanner !), plateforme en ligne conçue pour détecter automatiquement les pièges de modélisation fréquemment rencontrés lors du développement d'ontologies. Ces pièges (ou pitfalls) sont classés en plusieurs catégories, selon leur impact (important, moyen, faible) sur la qualité du modèle ;
— **OntoMetrics** plateforme d'analyse statistique qui calcule des indicateurs structurels tels que : le nombre de classes, propriétés et axiomes ; la profondeur moyenne des hiérarchies ; le taux de couverture (proportion de classes instanciées) ; la densité des relations entre concepts ;

Les raisonneurs HermiT et Pellet, très largement utilisés dans la communauté, se sont révélés particulièrement efficaces pour identifier des erreurs de modélisation difficilement repérables manuellement. Par ailleurs, les plugins associés à Protégé ont permis d'intégrer cette vérification tout au long du processus de modélisation.

Enfin, l'outil OOPS ! a été mobilisé à l'issue de la construction de l'ontologie, afin de réaliser une analyse plus systématique des erreurs structurelles potentielles.

5.2.2 Résultats de la validation structurelle

Dans le cadre de la construction et de la formalisation de l'ontologie sous Protégé, des outils de raisonnement tels que HermiT et Pellet ont été mobilisés à chaque étape du processus. Leur utilisation a permis de détecter rapidement d'éventuelles incohérences logiques, facilitant ainsi leur correction au fur et à mesure de l'avancement du modèle. En complément de cette

vérification continue, une analyse finale a été réalisée à l'aide de l'outil OOPS ! (Poveda-Villalon et al., 2014), afin d'identifier d'éventuelles erreurs conceptuelles ou de mauvaises pratiques non couvertes par les moteurs de raisonnement. Les résultats issus de cette analyse sont synthétisés dans le tableau 5.1.

Tableau 5.1 Synthèse de l'analyse du scanner OOPS !

Écueil	Occurrence	Gravité
Domain ou *Range* manquant dans les propriétés	120	Importante
Relation inverse non déclarée	4o	Mineure
Annotations manquantes	206	Mineure
Utilisation d'une classe "divers" (*Miscellaneous*)	2	Mineure
Utilisation d'une convention de nommage hétérogène	-	Mineure

Ils existent trois degrés d'importance pour ces écueils dans OOPS ! décrits de la manière suivante :

— Critique (*Critical*) : Il est essentiel de corriger cette lacune, car elle pourrait nuire à la cohérence de l'ontologie, au raisonnement ou à son applicabilité ;

— Important (*Important*) : Même si cette lacune n'affecte pas directement le fonctionnement de l'ontologie, il reste important de la corriger ;

— Mineure (*Minor*) : Ce n'est pas un véritable problème, mais sa correction permettrait d'améliorer la clarté ou la qualité de l'ontologie ;

Nous pouvons déjà observer qu'OntoIMR ne tombe dans aucun écueil critique, ce qui empêcherait son bon fonctionnement. Les trois écueils détectés dans l'ontologie font l'objet d'une discussion détaillée dans les sections suivantes.

5.2.3 *Domain* ou *Range* manquant dans les propriétés

L'analyse avec OOPS ! a relevé l'absence de *Domain* ou *Range* sur certaines propriétés (d'objet ou de données) de l'ontologie. D'un point de vue théorique, OWL recommande de définir ces contraintes afin de renforcer la sémantique et de faciliter la détection d'éventuelles incohérences. Cette situation, bien que signalée comme une anomalie importante, résulte d'un choix délibéré. En effet, dans le contexte de notre modèle pivot ERP-MES, certaines propriétés, comme *Name*, doivent pouvoir s'appliquer à des classes variées sans générer de contraintes d'inférence trop rigides. Dans le langage OWL DL, la définition de plusieurs *Domains* est interprétée par les raisonneurs de la façon suivante : si un élément est associé à une propriété ayant plusieurs classes pour *Domain* alors il appartient à l'intersection de toutes ces classes. Cette interprétation ne convient donc pas pour certaines de nos propriétés,il a été décidé

de ne pas spécifier ces axiomes. De même, dans le cas des *Object properties* définir uniquement la portée de la relation et non son domaine offre plus de flexibilité car plus de classe peuvent utiliser cette relation sans incohérence. Ce compromis offre une meilleure flexibilité pour intégrer des données issues de systèmes industriels hétérogènes et permet de réduire le risque d'erreurs de typage lors de la réutilisation ou de l'extension du modèle. Cette décision est donc considérée comme acceptable et cohérente avec les objectifs d'interopérabilité de l'ontologie.

5.2.4 Relation inverse non déclarée

Cet écueil apparaît lorsqu'une relation (sauf si définie comme une propriété symétrique avec owl :SymmetricProperty) n'a pas de relation inverse (owl : inverseOf) définie dans l'ontologie. Dans notre modèle, peu de relations inverse ont été définies afin de conserver une simplicité et clarté du modèle. Nous pourrions ajouter les relations inverses afin de maximiser la couverture logique et sémantique de l'ontologie. Par exemple, nous n'avons défini ici que la propriété d'objet : *usesResources* qui lie la classe *Process Segment* à la superclasse *Resource*, nous pourrions ajouter son inverse *ResourceIsUsedBy*. Cet écueil reste cependant d'importance mineure.

5.2.5 Annotation manquante

OOPS ! signale l'absence d'annotations pour certains éléments de l'ontologie, notamment de labels (rdfs :label) ou de descriptions (rdfs :comment). Cet écueil, considéré comme mineur, n'affecte pas la cohérence logique du modèle. En effet, il signifie que certaines classes ou propriétés n'ont pas été bien documentées, ce qui réduit la lisibilité pour des utilisateurs externes ou lors de futures réutilisations. Bien que cela n'entrave pas le fonctionnement opérationnel de l'ontologie, nous allons ajouter des descriptions aux classes et aux propriétés. L'objectif est de rendre le modèle aussi utilisable et réutilisable que possible.

5.2.5.1 Classe générique *Miscellaneous*

OOPS ! signale un écueil mineur lié à la présence d'une classe générique, considérée comme une mauvaise pratique car elle regroupe des éléments sans réelle cohérence sémantique. Dans notre cas, cet avertissement concerne la classe *PRT_Miscellaneous*, issue de notre modèle de ressources de production. Elle permet de représenter certains outils atypiques ne pouvant être classés ailleurs à ce stade. Son utilisation reste limitée et justifiée dans le contexte du modèle.

5.2.5.2 Conventions de nommage hétérogènes

Un écueil mineur identifié concerne l'usage de conventions de nommage non uniformes au sein de l'ontologie (par exemple l'alternance entre *CamelCase* et des noms contenant des underscores). Ce choix est toutefois volontaire : il permet, dans certaines *data properties*, d'indiquer explicitement les classes associées, lorsque cela est pertinent. Cette décision, guidée par des considérations de traçabilité et de lisibilité métier, ne sera donc pas modifiée.

5.2.6 Conclusion

La validation structurelle a été réalisée tout au long du développement à l'aide de raisonneurs logiques. Ensuite, l'utilisation de OOPS !, un outil de détection d'erreurs systématiques fréquentes, a permis de vérifier la bonne construction de l'ontologie. L'ensemble de ces vérifications n'a pas montré d'erreur critique et a permis d'identifier quelques ajustements mineurs destinés à renforcer la qualité du modèle. Cette vérification structurelle accomplie, nous pouvons désormais commencer la validation sémantique du modèle.

5.3 Validation sémantique

Alors que la validation structurelle s'intéresse à vérifier la fonctionnalité de l'ontologie quel que soit le domaine d'application, la validation sémantique vise à évaluer sa capacité à représenter fidèlement un domaine réel. Cette étape est essentielle pour vérifier que l'ontologie développée est effectivement en mesure de modéliser les gammes de fabrication telles qu'elles existent dans les systèmes industriels, et qu'elle peut répondre à des interrogations métiers concrètes. En d'autres termes, il ne suffit pas qu'une ontologie soit correctement construite sur le plan logique : elle doit également être utile, compréhensible et exploitable dans le cadre de cas d'usage spécifiques.

5.3.1 Données utilisées pour la validation sémantique

Afin de réaliser la validation sémantique, deux études de cas industrielles représentatives de la réalité terrain ont été utilisées. Elles permettent de s'assurer que les concepts utilisés dans les environnements ERP et MES sont bien couverts par le modèle ontologique proposé, renforçant ainsi sa pertinence en tant que modèle d'interopérabilité entre systèmes hétérogènes. Les deux cas d'étude exploités sont les suivants :

— Étude de cas ERP : extraite du logiciel ERP e-Prelude, cette gamme de fabrication

décrit la production d'une bibliothèque (*bookcase*). Les champs de données associées aux données des gammes de fabrication ont été extraite depuis les interfaces de configuration du logiciels ;

— Étude de cas MES : issue du logiciel MES SmartPortal, cette gamme est associée à la fabrication d'un vélo. Les champs de données et ses données associées ont été exportées sous format SQL ;

Pour chaque étude de cas, tous les champs présents et donc potentiellement remplissables dans les fichiers ont été analysés, qu'ils soient renseignés ou non. Chacun d'eux a été comparé aux concepts de l'ontologie afin de déterminer s'il existait une correspondance directe ou indirecte.

5.3.2 Résultats de la validation sémantique

Le tableau suivant 5.2 synthétise les résultats obtenus pour les deux cas d'étude, en calculant le taux de couverture sémantique. Pour chaque cas d'étude, chacun des champs de données a été étudié pour déterminer s'il correspondait à un concept dans l'ontologie. Le nombre de champs représentés dans l'ontologie a ensuite été rapporté au nombre total de champs présents (y compris ceux sans valeur renseignée) pour représenter la gamme dans chacun des cas d'étude.

Tableau 5.2 Taux de couverture de l'ontologie

Cas d'étude	Nombre de champs dans le cas d'étude	Nombre de champs représentés dans l'ontologie	Taux de couverture
e-Prelude	101	92	91,1%
SmartPortal	181	145	80,1%

L'ontologie a permis de représenter 91,1 % des champs extraits du système ERP et 78,5% des champs extraits du système MES. Ces résultats confirment la robustesse du modèle ontologique et sa capacité à représenter les concepts de différents logiciels avec cependant une meilleure représentation de la gamme issue de l'ERP. Les différents types de correspondances que nous avons rencontrés sont décrits dans la partie suivante 5.3.2.1, Cependant, nous n'atteignons pas un taux de couverture de 100%, il existe donc certaines limites de représentation que nous décrirons en 5.3.2.3

5.3.2.1 Typologie des correspondances observées

3 grandes catégories de correspondances ont été identifiées entre les champs métiers des cas d'étude et les concepts de l'ontologie :

— Correspondance directe : le nom (ou sa traduction) et la signification du champ sont strictement identiques à une classe ou propriété de l'ontologie (exemple le champs *bom_id* sur SmartPortal correspond au champ *BOM_ID* de notre ontologie) ;
— Correspondance sémantique équivalente : le champ porte un nom différent, mais son sens correspond clairement à un concept existant dans l'ontologie (par exemple, sur e-prelude, le nom de la majorité des concepts est intitulé *Libellé* ce qui correspond dans notre ontologie à la Data property *Name*) ;
— Correspondance indirecte : le champ était représenté dans ou pour une autre classe ou structure de l'ontologie, mais il peut être utilisé pour un autre concept d'un cas d'étude sans modification. (par exemple, en définissant notre modèle commun puis notre ontologie, le concept de *Scrapfixed* avait été défini originellement uniquement pour la classe *ProcessSegment*. Or dans la gamme de fabrication d'e-Prelude, le champs de *Rebut fixe* est aussi utilisé dans la Gestion des liens de nomenclature. Nous pouvons étendre sans modification de l'ontologie) ;

5.3.2.2 Champs exclus de la validation

Certains champs identifiés dans les cas d'étude ont été exclus du taux de couverture pour les raisons suivantes :

— Champs calculés automatiquement à partir d'autres données : des champs tels que le *niveau de nomenclature* sont générés dynamiquement à partir de relations existantes (ex. hiérarchie de composants) et ne sont pas configurables par l'utilisateur. Leur contenu est dérivé, ce qui les rend redondants au regard des concepts déjà couverts par l'ontologie ;
— Champs techniques internes aux logiciels : certaines structures, comme les champs au format *JSON* observés dans SmartPortal, contiennent des données spécifiques au fonctionnement ou à la configuration logicielle. Leur format non structuré et leur variabilité contextuelle ne permettent pas une modélisation générique et pertinente dans le cadre de l'ontologie ;
— Champs liés à la présentation ou à la précision d'affichage : ces champs définissent uniquement la mise en forme des données, sans portée métier. Par exemple, le champ *Décimale* dans e-Prelude indique le nombre de chiffres après la virgule pour l'affichage, sans impact sur la structure ou le sens des données ;

Ils représentent 5 % des champs de e-Prelude (10 champs) et 7% de SmartPortal (8 champs). L'exclusion de ces champs ne semble pas compromettre la capacité de l'ontologie à représenter de manière fidèle les concepts métiers centraux liés aux gammes de fabrication. En effet, les éléments écartés sont principalement des données techniques, calculées ou de présentation,

qui ne relèvent pas directement de la logique métier modélisée. Ce choix méthodologique vise également à préserver la généricité et la maintenabilité de l'ontologie, en se concentrant sur des concepts réutilisables dans différents contextes industriels. Il reste toutefois possible, dans des applications spécifiques, d'enrichir ultérieurement le modèle pour inclure certains de ces champs si un besoin fonctionnel clair le justifie.

5.3.2.3 Description des limites de représentation observées

Malgré un haut taux de couverture, certains champs ne sont pas représentés. Nous les avons identifiés comme faisant partie des catégories suivantes :

— Champs très spécifiques au contexte d'utilisation ou peu documentés : certains champs sont fortement liés à des contextes industriels particuliers (ex. métallurgie, agroalimentaire) ou n'ont pas pu être interprétés de manière fiable en l'absence de documentation suffisante. Par précaution méthodologique, ils n'ont pas été intégrés. Exemples : ,*alloy_id*, *treatment_id*, etc. Ces champs pourraient être réévalués et modélisés dans le cadre d'une spécialisation sectorielle ou d'une entreprise spécifique de l'ontologie ;

— Champs liés à la traçabilité, aux événements et à l'exécution : ces champs sont utilisés pour suivre l'exécution des opérations en temps réel, enregistrer des événements ou assurer la traçabilité des lots et des produits. Bien qu'ils ne soient pas essentiels à la description structurelle d'une gamme, ils pourraient s'avérer pertinents dans une ontologie orientée vers l'exécution ou le pilotage MES. Exemples : *trace_batch*, *rt_evt_id*, *rt_shift_id*, etc. Une extension future pourrait intégrer ce type d'information dans une perspective plus opérationnelle ;

— Champs de configuration ou de gestion organisationnelle : certains champs permettent d'associer des paramètres de gestion ou des responsabilités humaines (ex. gestionnaire, mode de déclaration), mais ne décrivent pas directement le déroulement technique des opérations. Exemples : *Déclaration (point de comptage)*, *Gestionnaire*. Ces éléments pourraient enrichir l'ontologie si l'on visait une intégration complète avec des modèles organisationnels ou des logiques de supervision ;

— Champs techniques internes ou de contrôle d'interface : ces champs, souvent générés automatiquement par les systèmes ou liés à la logique de formulaire, sont utilisés pour le fonctionnement interne ou l'affichage dynamique des interfaces. Leur rôle est généralement purement technique et dépendant de l'implémentation logicielle, ce qui limite leur réutilisabilité dans un cadre générique. Exemples : *val_usr_id*, *ctrl_res_id*, *ctrl_line_step*, *ctrl_line_order*, etc. Cependant, dans un contexte d'intégration technique plus poussé, certains de ces champs pourraient justifier une modélisation complémentaire ;

L'exclusion actuelle de ces champs ne remet pas en cause la capacité de l'ontologie à représenter de manière cohérente et fonctionnelle les éléments essentiels d'une gamme de fabrication. Toutefois, ces catégories pourraient faire l'objet d'un enrichissement ultérieur si l'objectif évolue vers une couverture plus fine des aspects d'exécution, de traçabilité ou de pilotage opérationnel.

5.3.3 Conclusion

Les résultats obtenus démontrent que l'ontologie conçue est générique, réutilisable et apte à modéliser des gammes de fabrication issues de systèmes industriels différents. Le taux de couverture élevé confirme la capacité du modèle à représenter l'essentiel des concepts industriels utilisés en pratique dans les ERP et MES.

5.4 Preuves de concept

Les sections précédentes ont démontré que notre modèle ontologique permet de représenter avec précision et cohérence les gammes de fabrication, qu'elles proviennent d'un ERP ou d'un MES. Toutefois, pour valider pleinement le potentiel de ce modèle dans un contexte d'interopérabilité industrielle, il est essentiel de tester sa capacité à servir de pont entre des systèmes hétérogènes comme indiqué sur le schéma 4.1. C'est dans cette optique que deux preuves de concept ont été réalisées. Ces expérimentations visent à démontrer la faisabilité technique de deux opérations complémentaires :

- Alimenter l'ontologie à partir des données d'un ERP ou d'un MES existant (intégration ascendante) ;
- Utiliser les données contenues dans l'ontologie pour renseigner automatiquement un autre système (intégration descendante) ;

Le mécanisme vertical idéal s'appuie sur des web services, cependant nous n'avions pas les accès logiciels nécessaires dans le temps imparti pour le faire, c'est pourquoi nous présentons pour nos deux preuves de concepts les sources de données utilisées et les modes de communication déployés pour montrer que la communication par l'ontologie est possible sans développer de web services.

5.4.1 PoC1 — Intégration depuis les logiciels ERP et MES vers l'ontologie

Cette première preuve de concept a consisté à valider la capacité du modèle ontologique à intégrer automatiquement des données réelles issues d'un système ERP et d'un système MES. L'objectif était de vérifier que l'ontologie pouvait être alimentée sans intervention manuelle à

partir de formats industriels standards, tout en respectant sa structure logique et sémantique. C'est-à-dire vérifier la partie de communication verticale illustrée sur la figure 5.1.

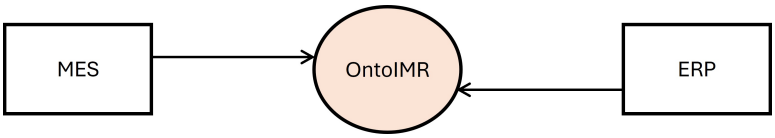

FIGURE 5.1 Schéma de communication verticale partielle, des logiciels vers OntoIMR

Démonstrateur pour la PoC1

Dans un contexte industriel réel, l'interopérabilité entre un ERP et un MES repose généralement sur l'utilisation de services web, d'API ou de connecteurs standards permettant une communication automatique, sécurisée et bidirectionnelle entre les systèmes. Toutefois, dans le cadre de ce projet, il n'a pas été possible de mettre en place une telle intégration complète. Cette limitation s'explique principalement par l'absence d'accès direct aux environnements logiciels des entreprises partenaires, le temps restreint imparti au développement, ainsi que par le manque de ressources techniques nécessaires pour configurer une infrastructure de communication stable et sécurisée (droits d'administration, serveurs intermédiaires, environnements de test isolés, etc.).

En conséquence, la preuve de concept présentée ici repose sur un démonstrateur basé sur les deux sources distinctes suivantes :

— Un export Excel du système ERP *e-Prelude*, contenant les informations de gamme de fabrication : opérations, séquences, temps associés, ressources mobilisées, etc ;
— Une extraction SQL du MES *SmartPortal* exporté en Excel, décrivant les routes de fabrication d'un produit (séquences, opérations, taux, documents, etc.) ;

Ces deux jeux de données représentent des structures différentes, mais partagent une sémantique industrielle commune autour de la notion de gamme de fabrication, ce qui rend leur intégration dans une ontologie pivot particulièrement pertinente. Le schéma 5.2 reprend le schéma de communication verticale 5.1 avec les sources de données décrites précédemment.

En plus de ces deux jeux de données, un fichier Excel de mapping a également été utilisé. Celui-ci faisait le lien entre les champs des sources de données (noms de colonnes ERP/MES) et les entités correspondantes dans l'ontologie (classes cibles, types de propriétés, noms de relations, etc.).

Bien que cette approche dégrade le niveau d'automatisation attendu dans un contexte opérationnel, elle permet de tester la validité du modèle ontologique dans un flux de communication

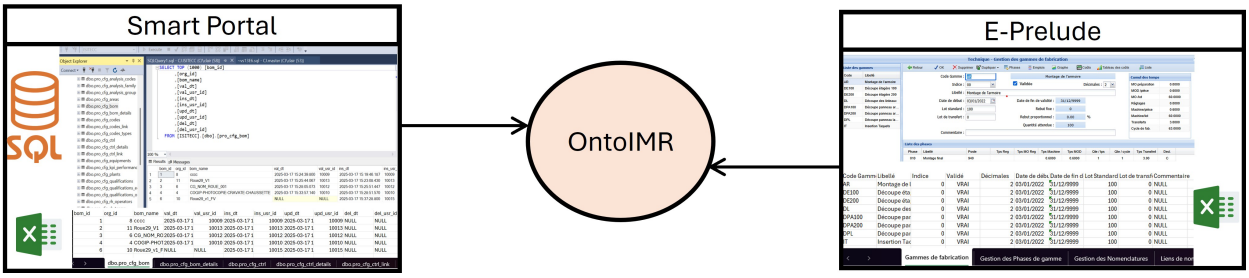

FIGURE 5.2 Schéma des sources des données pour la communication verticale partielle

ascendant, en simulant l'alimentation de l'ontologie à partir de données issues d'un ERP ou d'un MES réel. Ce compromis méthodologique permet ainsi de valider la capacité du modèle à absorber et structurer des données hétérogènes, en s'inscrivant dans une logique de préfiguration des échanges automatisés.

Méthodologie de traitement

Un script Python a été développé pour transformer automatiquement ces données en individus OWL, conformément à la structure définie dans notre ontologie. Ce script :

- lit les fichiers d'entrée (Excel) ;
- associe chaque champ à une classe ou propriété de l'ontologie via le fichier de mapping ;
- crée des individus OWL en instanciant les bonnes classes et en reliant correctement les entités entre elles (ex. : séquence liée à une opération, ressource à une opération, etc.) ;
- exporte un fichier OWL prêt à être importé dans Protégé ;

Le développement de ce script s'est appuyé sur des outils d'intelligence artificielle afin de garantir la conformité syntaxique du code généré, d'automatiser le traitement des correspondances, et de faciliter les itérations.

Résultats obtenus

L'ontologie a pu être alimentée sans erreur à partir des données issues de e-Prélude et Smart-Portal, les résultats pour ces deux validations sont visibles respectivement à la figure 5.3 et 5.4. Les individus générés sont correctement typés, reliés entre eux selon les relations prévues (par exemple : *hasOperation*, *usesResource*, *hasNextSequence*, etc.), et validés via le raisonneur logique *Pellet*.

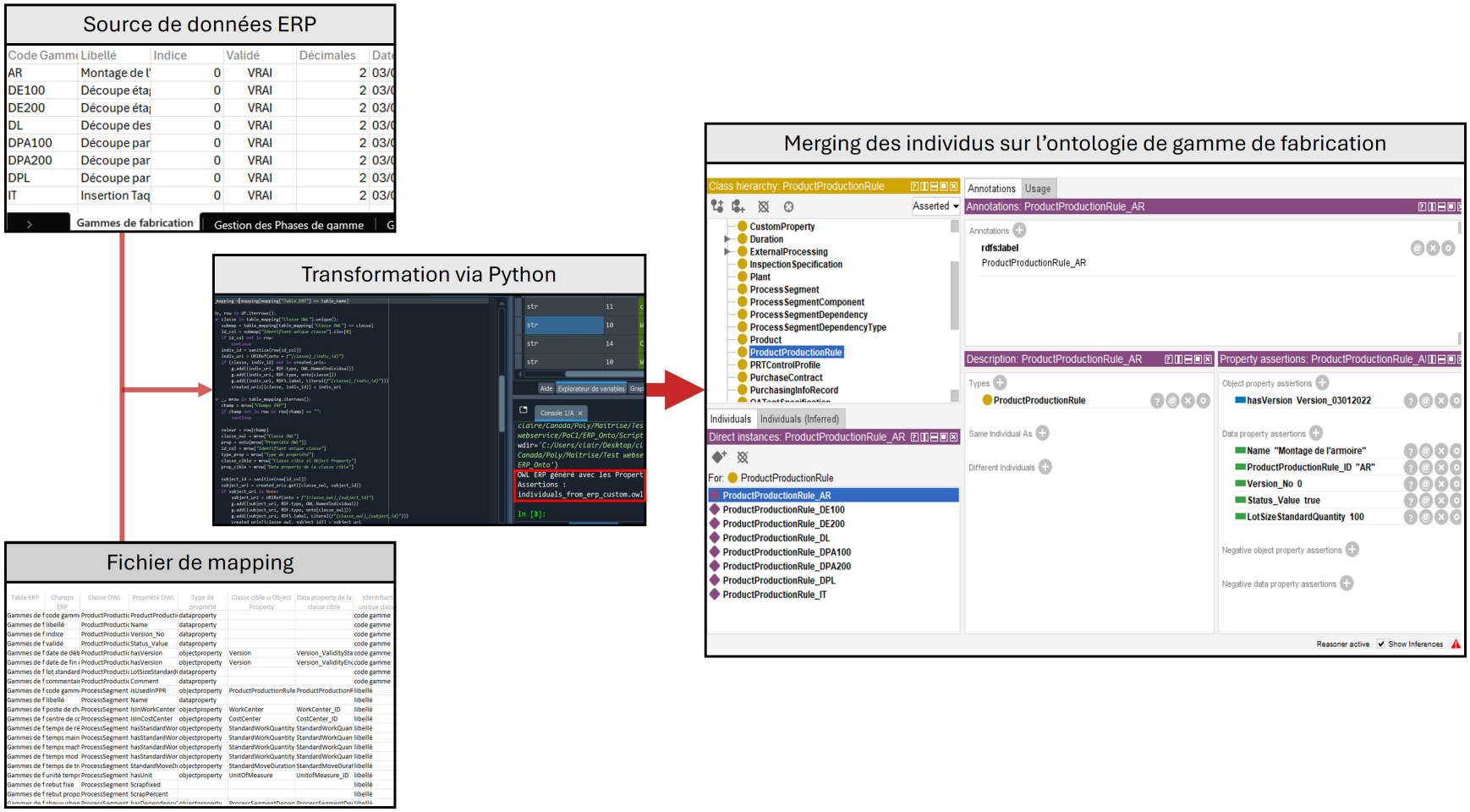

FIGURE 5.3 Preuves de concept 1 pour e-prélude

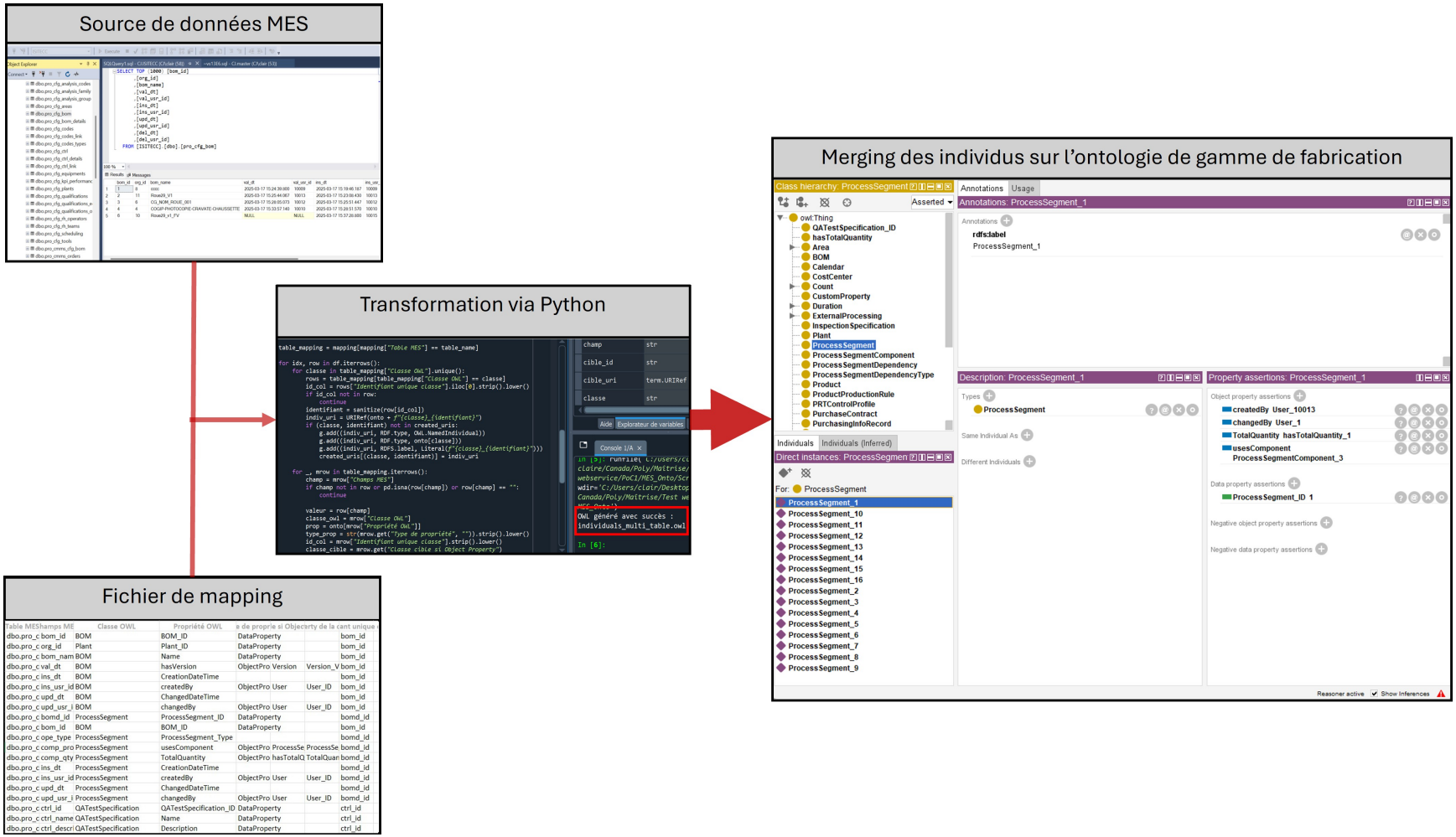

FIGURE 5.4 Preuves de concept 1 pour SmartPortal

Bien que cette preuve de concept ait permis de tester avec succès l'intégration ascendante de données vers l'ontologie, certaines limites doivent être prises en compte. Les données exploitées ont été extraites manuellement sous forme de fichiers Excel à partir des systèmes ERP et MES, sans recours à une communication dynamique via API ou service web, ce qui ne reflète pas encore un flux de données en temps réel. Le processus repose également sur un fichier de mapping externe, indépendant de l'ontologie, dont la maintenance devra être assurée en parallèle. Par ailleurs, le script suppose des fichiers bien structurés et ne gère pas encore les cas d'erreurs, de doublons ou de données manquantes. En outre, seule une partie des concepts de l'ontologie a pu être instanciée, en fonction des informations réellement disponibles dans les systèmes testés. Enfin, la validation s'est concentrée sur la cohérence logique des données (via un raisonneur OWL), sans mise en situation fonctionnelle dans les logiciels sources.

Malgré ces limites, cette première expérimentation a permis de démontrer qu'OntoIMR est capable :

- d'absorber des données provenant de logiciels hétérogènes ;
- de réconcilier ces données autour d'une structure pivot cohérente ;
- de préparer les données pour une éventuelle réutilisation descendante dans d'autres systèmes ;

Elle confirme donc la pertinence de l'ontologie comme structure commune pour unifier les gammes de fabrication provenant de systèmes industriels distincts. Fort de cette première validation ascendante, la deuxième preuve de concept s'intéresse maintenant à la capacité du modèle à redistribuer ces données vers un système cible, illustrant ainsi la faisabilité d'une communication descendante.

5.4.2 PoC2 — Intégration depuis l'ontologie vers la base de données du MES SmartPortal

Cette seconde preuve de concept complète le flux de communication verticale initié dans la PoC1, en validant cette fois la capacité à extraire des données depuis l'ontologie pour alimenter automatiquement la base de données d'un MES ou d'un ERP. C'est-à-dire la partie du mécanisme idéal de la figure 4.1, illustrée à la figure 5.5 Dans notre cas, nous n'avons accès qu'à la base de données SQL du MES, nous ne nous intéresserons qu'à la transmission d'information de l'ontologie vers le MES mais celle de l'ontologie vers une base de données ERP est très similaire.

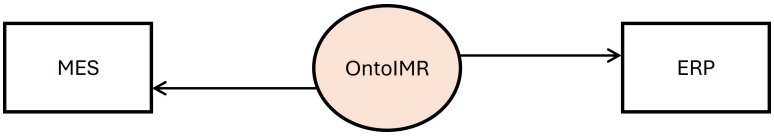

FIGURE 5.5 Schéma de communication verticale partielle, d'OntoIMR vers les logiciels

Démonstrateur pour la PoC2

Comme évoqué précédemment, l'idéal en contexte industriel serait de transmettre automatiquement les données de l'ontologie vers un système MES à l'aide d'un service web ou d'une API dédiée. Toutefois, compte tenu des contraintes déjà mentionnées dans la PoC1 (accès restreint aux environnements logiciels, délais limités, absence d'infrastructure d'intégration disponible), la preuve de concept a été réalisée via une approche alternative, reposant sur l'export des données ontologiques sous forme de fichiers intermédiaires.

Ce démonstrateur s'appuie sur les individus OWL générés à partir des données de gamme extraites de l'ERP e-Prelude, intégrés dans l'ontologie lors de la PoC1. Ces individus représentent des entités telles que les opérations, les séquences, les ressources ou encore les temps opératoires, toutes reliées selon la structure logique définie par OntoIMR. La figure 5.6 illustre les sources mobilisées pour cette communication simulée de l'ontologie vers la base de données du MES SmartPortal.

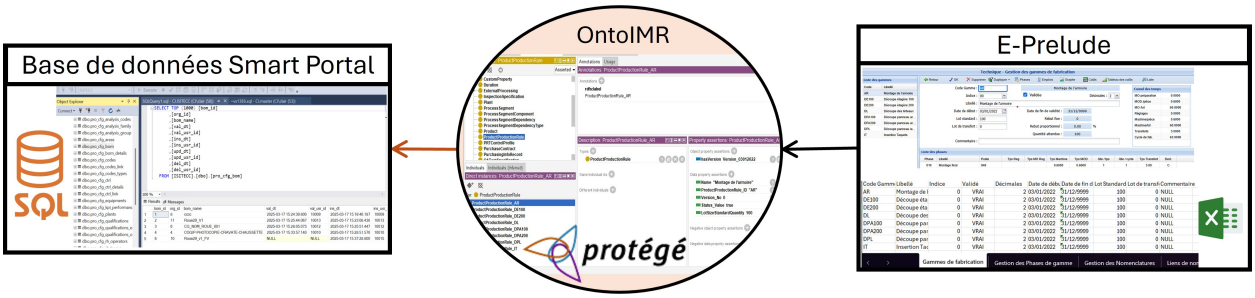

FIGURE 5.6 Schéma des sources des données pour la communication verticale partielle

Méthodologie de traitement

Un script Python a été développé pour effectuer les opérations suivantes :

- Lire le fichier OWL contenant les individus issus d'e-Prelude ;
- Identifier les classes et propriétés pertinentes pour la base de données du MES SmartPortal (ex. : *Operation*, *Sequence*, *usesResource*, *hasTime*) ;

— Extraire les données (nom, durée, ordre, type de ressource, taux, etc.) selon la structure définie dans l'ontologie ;
— Générer un fichier CSV structuré ou un script SQL correspondant à la structure attendue par la base de données du MES ;

La structure de correspondance entre les concepts OWL et les tables SQL du MES a été définie à l'aide du même fichier de mapping utilisé pour la PoC1.

Les données sont ensuite injectées à l'aide de requêtes SQL dans la base de données du MES sur SQL Server Management Studio Pro 20.

Résultats obtenus

Les données issues de l'ERP e-Prelude, structurées dans l'ontologie, ont pu être transformées sans erreur en lignes SQL compatibles avec les tables du MES SmartPortal.

Les informations sur les opérations, les séquences, les durées et les ressources ont été insérées correctement dans les tables correspondantes. Les relations entre entités (par exemple : liens entre séquence et opération, ou entre opération et ressource) ont également été respectées, conformément au modèle d'origine.

Les résultats de cette preuve de concept sont illustrés en figure 5.7, qui présente un extrait d'une capture d'écran du fichier CSV généré pour alimenter la base de données adaptée à la structure SQL.

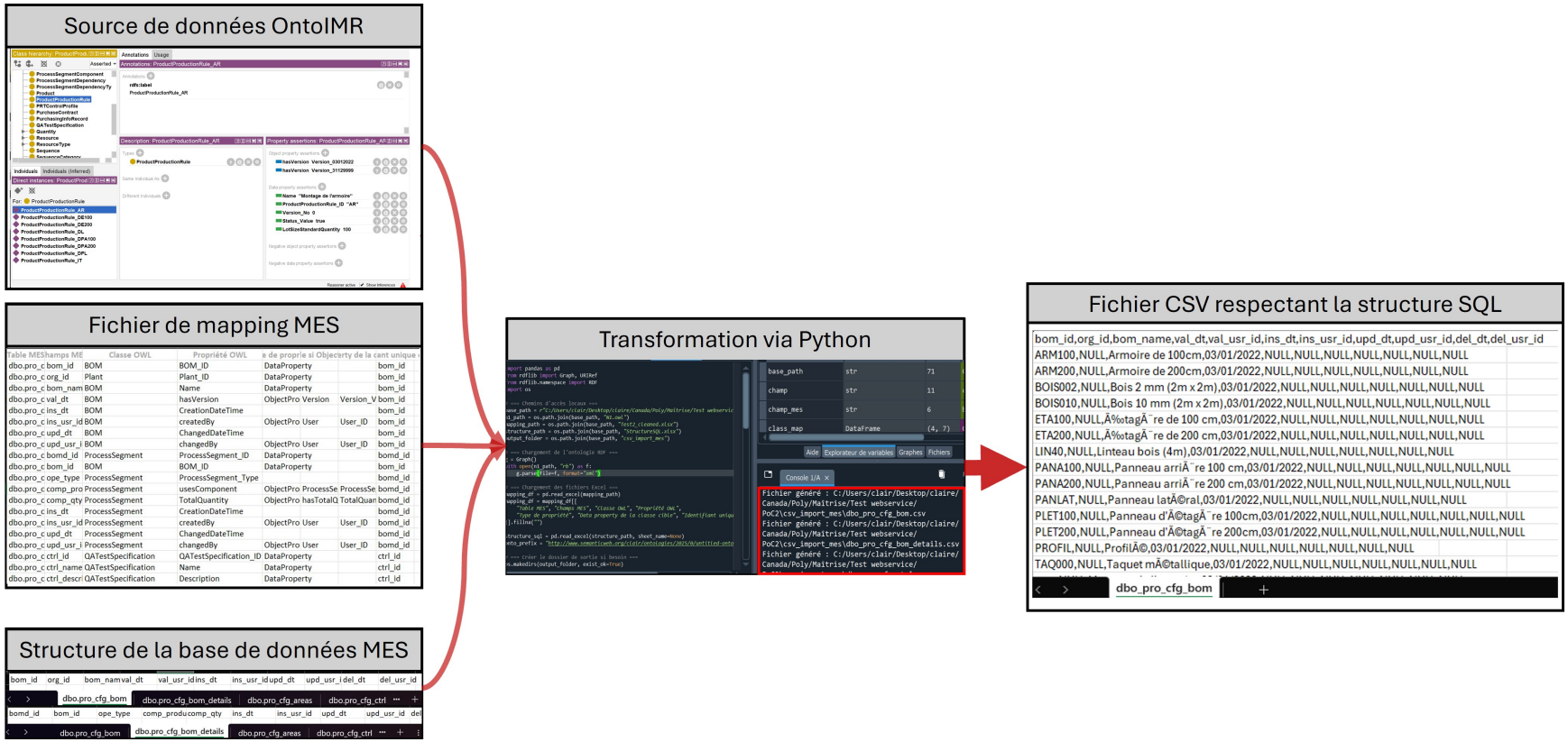

FIGURE 5.7 Preuve de concept 2

Cette seconde preuve de concept, bien qu'elle complète le flux de communication verticale amorcé dans la PoC1, présente certaines limites. Les données n'ont pas été injectées directement dans la base SQL du MES SmartPortal, mais seulement préparées sous forme de fichiers CSV ou de scripts SQL structurés selon les attentes du schéma relationnel. Il n'y a donc pas eu de test d'intégration complet ni de validation fonctionnelle au sein du logiciel MES. Par ailleurs, l'export repose sur un mapping défini manuellement entre les concepts OWL et les tables SQL, ce qui nécessite une adaptation spécifique pour chaque structure de base cible. Le script n'intègre pas de mécanisme de vérification métier ou de retour d'erreur depuis le système récepteur. Enfin, seules certaines entités présentes dans l'ERP ont été transformées, ce qui ne permet pas encore de valider l'ensemble du spectre couvert par l'ontologie.

Malgré ces limites, cette preuve de concept permet ainsi de démontrer qu'OntoIMR est capable :

- de restituer des données précédemment intégrées dans l'ontologie de manière structurée et exploitable ;
- de transformer automatiquement ces données selon les exigences d'un système MES, ici SmartPortal ;
- d'alimenter la base de données d'un MES via un export SQL, à partir d'informations initialement issues d'un ERP (e-Prelude), illustrant ainsi un flux de communication verticale complet ;

Conclusion sur les preuves de concept

Les deux preuves de concept réalisées confirment la capacité de l'ontologie OntoIMR à assurer la communication verticale entre un ERP et un MES, dans les deux sens.

La première preuve de concept (PoC1) a validé l'intégration ascendante : des données issues de l'ERP e-Prelude et du MES SmartPortal ont pu être automatiquement converties en individus OWL, en respectant la structure logique de l'ontologie. Cela démontre la capacité d'OntoIMR à absorber des données hétérogènes et à les centraliser autour d'un modèle pivot cohérent.

La seconde preuve de concept (PoC2) a démontré l'intégration descendante : les données, une fois modélisées dans l'ontologie, ont pu être restituées sous forme de fichiers structurés prêts à être insérés dans la base de données du MES à l'aide de requêtes SQL. Cette opération prouve qu'OntoIMR peut aussi préparer et redistribuer des données vers d'autres systèmes industriels.

Ainsi, OntoIMR remplit pleinement son rôle de couche sémantique d'interopérabilité, en

assurant un échange bidirectionnel cohérent de données de fabrication entre environnements logiciels distincts.

5.5 Conclusion

La validation menée a permis de confirmer la robustesse de l'ontologie selon trois axes complémentaires. D'abord, la validation structurelle a assuré la cohérence logique du modèle et sa conformité aux standards OWL. Ensuite, la validation sémantique, fondée sur deux cas d'étude réels, a démontré sa capacité à représenter fidèlement les concepts des gammes de fabrication issues d'ERP et de MES. Enfin, les preuves de concept ont validé l'interopérabilité effective de l'ontologie en illustrant son aptitude à intégrer et restituer des données entre systèmes hétérogènes.

Ces résultats ouvrent la voie à la dernière partie du mémoire, qui propose une synthèse des travaux réalisés, en expose les limites, et suggère des pistes d'amélioration pour de futures recherches.

CHAPITRE 6 CONCLUSION

Dans l'industrie manufacturière, les systèmes ERP et MES sont essentiels pour gérer respectivement la planification des ressources et l'exécution des opérations. Bien qu'ils manipulent des objets métier communs — notamment les gammes de fabrication, qui décrivent l'enchaînement des opérations nécessaires à la production d'un article — ces deux systèmes ne reposent pas sur un modèle de données unifié. Ce manque d'harmonisation engendre des saisies redondantes, des risques d'erreurs de synchronisation, et une perte d'efficacité lors des mises à jour entre les niveaux de planification et d'exécution.

Dans un contexte de transformation numérique accélérée et de diffusion des principes de l'Industrie 4.0, renforcer l'interopérabilité entre ERP et MES est devenu un enjeu stratégique, tant pour améliorer la cohérence des systèmes d'information que pour automatiser les échanges de données à travers toute la chaîne de valeur.

6.1 Synthèse des travaux / Summary of Works

Ce mémoire s'inscrit dans la problématique d'interopérabilité entre ERP et MES, en ciblant un objet métier central : la gamme de fabrication. Bien que présente dans les deux systèmes, cette dernière est souvent modélisée de manière hétérogène, ce qui freine la synchronisation des données entre les niveaux de planification et d'exécution.

Afin de répondre à cette problématique, nous avons conçu un modèle ontologique interopérable, destiné à représenter de manière unifiée et explicite les gammes de fabrication dans un environnement industriel. Ce modèle vise à servir de structure pivot facilitant l'échange de données entre logiciels ERP et MES, tout en assurant leur cohérence sémantique. Il répond également à un enjeu important du point de vue du MES, où la valeur ne réside pas uniquement dans la transmission de la gamme de fabrication dans son ensemble, mais aussi dans la précision et la richesse des informations associées à chacune de ses opérations. En effet, la gamme n'est pas qu'une simple séquence d'étapes : chaque opération qui la compose porte des données critiques — paramètres d'exécution, ressources réellement mobilisées, documents associés, résultats d'inspection. En intégrant ces éléments opérationnels directement dans la structure de la gamme, la solution proposée permet, lors de son échange, de véhiculer non seulement la logique globale du processus, mais aussi les détails nécessaires à une exploitation par le MES, favorisant ainsi une intégration plus fine entre planification et exécution, et permettant une utilisation efficace de données en temps réel reçues par le MES.

La démarche a commencé par une analyse comparative approfondie de la manière dont plusieurs ERP (SAP, Business Central, Divalto, JobBoss) et MES (Sepasoft, SmartPortal) structurent les gammes de fabrication ainsi que par l'étude des recommandations issues de la norme ISA-95. Cette étape a permis de constater que la norme ISA-95 propose une base utile mais ne couvre pas de manière complète ni opérationnelle les modèles de gamme utilisés dans les logiciels actuels et que les implémentations logicielles observées varient fortement d'un éditeur à l'autre, et aucun modèle unifié n'est réellement utilisé.

À partir de ce constat, un modèle commun de gamme de fabrication a été élaboré, structurant les entités essentielles (opérations, séquences, ressources, temps, documents, composants, inspections, etc.) et leurs relations. Ce modèle a ensuite été formalisé sous forme d'ontologie à l'aide du langage OWL et de l'outil Protégé, offrant ainsi une base exploitable, interrogeable et extensible.

L'ontologie développée enrichit et étend donc la norme ISA-95 en intégrant plusieurs notions absentes ou peu formalisées dans la norme. OntoIMR couvre entre 80 et 90% (en fonction du type de logiciel) des éléments rencontrés dans les cas d'étude réels, en augmentant donc le taux de couverture de la norme ISA-95.

Le modèle a été validé selon trois axes complémentaires : par des tests de cohérence structurelle à l'aide de raisonneurs, par une validation sémantique sur deux cas d'usage réels (ERP et MES), et par la réalisation de preuves de concept démontrant sa capacité à assurer des échanges de données bidirectionnels.

En résumé, ce mémoire propose une approche novatrice pour améliorer l'interopérabilité dans le domaine des systèmes industriels, en introduisant un modèle ontologique dédié aux gammes de fabrication, formellement structuré, validé expérimentalement, et apte à soutenir des échanges bidirectionnels entre systèmes hétérogènes.

6.2 Limitations de la solution proposée / Limitations

Bien que les résultats obtenus soient encourageants, certaines limites doivent être soulignées.

D'abord, le nombre de logiciels analysés reste limité. Il ne couvre pas toutes les variantes possibles de structuration de gamme dans l'industrie, notamment celles utilisées dans des secteurs à forte spécialisation (aéronautique, pharmaceutique, agroalimentaire). Certains logiciels possèdent des modules spécifiques ou des champs métiers très ciblés qui ne sont pas pris en compte dans la version actuelle du modèle.

Ensuite, même si le modèle proposé enrichit la norme ISA-95, il ne couvre pas encore l'ensemble des concepts qu'un système MES ou ERP peut manipuler autour d'une gamme. Des

éléments périphériques comme la gestion de la qualité, la maintenance préventive, ou les ajustements de capacité en temps réel ne sont pas intégrés à cette étape. Preuve que la norme ISA-95 et OntoIMR peuvent encore être enrichies.

Par ailleurs, les preuves de concept ont été réalisées de manière déconnectée, via des fichiers Excel, faute d'accès direct aux API logicielles. Cela limite la validation technique à un contexte de test semi-automatisé, sans preuve de performance en temps réel ni intégration continue dans des systèmes en production.

6.3 Améliorations futures / Future Research

Les perspectives ouvertes par ce travail sont nombreuses et permettent d'imaginer plusieurs axes d'approfondissement.

Un premier prolongement naturel serait d'élargir la couverture fonctionnelle de l'ontologie. Cela inclurait l'intégration de concepts liés aux coûts, à des aspects de logistique non pris en compte (gestion des stocks, des entrepôts), à une gestion du personnel plus poussée (compétences, affectations), ou encore aux conditions environnementales (température, pression, humidité). Cela permettrait d'approcher une modélisation plus complète d'un processus de fabrication réel.

Il serait également pertinent de valider le modèle sur d'autres logiciels ERP et MES, en collaboration avec des industriels. Cela permettrait non seulement de tester la généricité du modèle dans d'autres contextes, mais aussi d'adapter certaines structures à des particularités sectorielles ou organisationnelles.

D'un point de vue technique, une étape importante consisterait à intégrer le modèle dans un environnement d'échange en temps réel, via des API REST, des web services SOAP, ou une architecture middleware (de type ESB ou iPaaS). Cela permettrait d'évaluer l'ontologie non plus seulement comme un modèle, mais comme une véritable couche d'intégration dynamique dans un SI manufacturier. Enfin, un dernier axe d'évolution intéressant serait le développement d'interfaces utilisateurs permettant de visualiser, manipuler et exploiter les données de l'ontologie par des acteurs métiers non spécialistes. Cela pourrait passer par des interfaces graphiques, des visualisations interactives ou des formulaires dynamiques connectés.

6.4 Conclusion générale

Ce mémoire apporte une contribution originale à la problématique d'interopérabilité entre ERP et MES, en proposant un modèle ontologique de gamme de fabrication comme solu-

tion pivot. Ce modèle permet d'unifier la structure des données, d'en faciliter l'échange et d'en renforcer la cohérence. En enrichissant la structure proposée par la norme ISA-95 et en démontrant sa faisabilité par des cas concrets, il constitue une base solide pour le développement de solutions d'intégration plus intelligentes et durables dans le cadre de l'Industrie 4.0.

ANNEXE A TABLEAU COMPARATIFS DES BASES DE DONNÉES

TABLEAU A.1 Comparaison des systèmes de gestion de données justification

Critères	**Hiérarchique**	**En réseau**	**Relationnel**	**Orienté objet**	**NoSQL**	**Ontologie**
Variété des solutions	Principalement des systèmes propriétaires. (Blatt, 1987)	Solutions propriétaires et OpenSource (Blatt, 1987)	Plateformes open source ou fermées. (Phiri et Kunda, 2017)	A la fois OpenSource et commerciales. (Suri et Sharma, 2011)	La majorité des systèmes NoSQL sont open source. (Phiri et Kunda, 2017)	Les frameworks d'ontologies peuvent être à la fois open source et propriétaires. (Sir et al., 2015)
Variété des solutions	-	+	+	+	+	+

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Scalabilité	Limité pour des données interconnectées, peut monter en puissance en améliorant le matériel d'un seul serveur.	Scalable avec des outils matures en améliorant le matériel.	Monte en puissance en améliorant le hardware d'un seul serveur. Scalable avec des outils matures, mais peut rencontrer des difficultés avec des données fortement interconnectées. (Phiri et Kunda, 2017)	Bien adaptées aux données complexes, mais limitées en scalabilité horizontale. Peuvent monter en puissance (verticalement) avec du matériel plus performant. (Ijmter, 2014)	Extensible horizontalement en distribuant les données sur différents serveurs. Très scalable et ce particulièrement avec un large volume de données non structurées. (Phiri et Kunda, 2017) , (Cattell, 2011)	Évolutif pour les requêtes complexes basées sur la connaissance, mais il peut s'avérer difficile de traiter de très grands volumes de données. (Staab et Studer, 2009)
Scalabilité	+	+	+	+	++	+

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Volume de données	Volumes modérés avec sa structure en arborescence. (Blatt, 1987)	Supporte de grands volumes grâce à sa structure plus complexe et interconnectée.(Blatt, 1987)	Gère des volumes de données limités. Gère bien les gros volumes dans des applications traditionnelles, mais peut éprouver des difficultés avec des données très volumineuses et fortement interconnectées. (Phiri et Kunda, 2017)	Peut gérer des larges volumes importants de données complexes (ex. : multimédia, objets hiérarchiques). Ils performent bien mieux que les bases de données relationnelles pour des données complexes et interreliées. (Ijmter, 2014)	Conçu pour gérer de grands volumes de données distribuées de manière efficace. (Phiri et Kunda, 2017) , (Cattell, 2011)	Gère efficacement des volumes moyens de données, mais ne sont pas optimisées pour de grands volumes de données. (Staab et Studer, 2009)
Volume de données	+	++	+	++	++	+

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Performance de requêtes	Les requêtes sont rapides si elles suivent la hiérarchie mais moins performantes pour des recherches complexes. (Suri et Sharma, 2011), (Blatt, 1987)	Offre de bonnes performances pour des requêtes complexes. (Blatt, 1987)	Peut devenir inefficace pour des requêtes complexes nécessitant plusieurs jointures. (Phiri et Kunda, 2017)	Rapides et efficace avec des requêtes et des structures de données. (Ijmter, 2014)	Excellente pour des requêtes distribuées et flexibles, mais peut rencontrer des difficultés avec des relations complexes. (Phiri et Kunda, 2017) , (Cattell, 2011)	Efficace pour des requêtes complexes et des tâches de raisonnement, performe d'autant mieux pour des requêtes impliquant des relations sémantiques. (Sir et al., 2015)
Performance de requêtes	-	++	+	++	++	++

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Complexité	Simples mais peut devenir complexes avec des relations de données rigides (Blatt, 1987)	Complexes à concevoir pour être adapté à des systèmes aux relations complexes. (Blatt, 1987)	Crée des données complexes lorsque les données à stocker par les utilisateurs sont difficiles à convertir en tables. (Phiri et Kunda, 2017) , (Suri et Sharma, 2011)	Plus complexes que les bases de données relationnelles en raison de la prise en charge de la hiérarchie des objets, des relations complexes et des opérations (méthodes) sur les données. (Suri et Sharma, 2011) , (Ijmter, 2014)	Moins complexes pour des opérations simples mais plus complexes lorsqu'il faut gérer des requêtes complexes et maintenir la cohérence des données complexes sont nécessaires. (Phiri et Kunda, 2017) , (Cattell, 2011)	Les ontologies sont complexes à modéliser, notamment pour représenter des concepts et relations fortement interconnectés (Staab et Studer, 2009), (Gruber, 1993)
Complexité	++	-	-	- -	-	- -

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Cohérence/ intégrité de données	Bonne cohérence mais nécessite une duplication de données dans certains cas. (Blatt, 1987)	Meilleure intégrité grâce à la flexibilité de la structure réseau, réduisant la redondance. (Blatt, 1987)	Repose sur des contraintes comme les clés étrangères et la normalisation pour maintenir l'intégrité.	L'intégrité est maintenue via l'encapsulation et l'héritage dans les modèles d'objets. (Ijmter, 2014)	Offre faible cohérence avec son approche sans schéma. Peut offrir une intégrité de données mais ne vient pas naturellement avec des méthode type ACID. (Phiri et Kunda, 2017), (Cattell, 2011)	Garantit une intégrité élevée grâce à la logique formelle et aux règles, mais manque de validation de contraintes dans les hypothèses de monde ouvert. (Sir et al., 2015), (Gruber, 1993)
Cohérence/ intégrité de données	+	+	++	+	- -	++

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Sécurité	Peu de contrôle granulaire. La sécurité est souvent assurée par des couches supplémentaires	Possibilité des contrôler plus précisément les accès grâce à la modularité des structures en réseau.	Modèles de sécurité matures avec prise en charge des rôles, des permissions et du chiffrement. (Phiri et Kunda, 2017)	Peut mettre en œuvre une sécurité plus granulaire, mais moins mature en termes d'outils (que le modèle relationnel). (Suri et Sharma, 2011)	La sécurité est laissée à la gestion par le middleware et n'est pas intrinsèque à la base de données. (Phiri et Kunda, 2017)	Intègre des mécanismes pour gérer l'accès et sécuriser les données à travers des vocabulaires contrôlés et des structures sémantiques. Il est souvent plus complexe a gérer que le modèle relationnel. (Sir et al., 2015)
Sécurité	+	+	++	+	-	+

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Gestion de type de données complexes	Mal adaptée pour des types de données complexes comme le multimédias. (Blatt, 1987)	Bonne gestion des types de données complexes, types multimédias. (Blatt, 1987)	Types de données simples (nombre, caractères, date et heure, binaire etc.).(Suri et Sharma, 2011)	Capable de gérer plusieurs types de données : multimédia (texte, nombres, images, des vidéos, voix ...). (Suri et Sharma, 2011), (Ijmter, 2014)	Peuvent stocker des types de données complexes, y compris des données multimédias et flexible (json et bson par exemple), des objets hiérarchiques et des structures imbriquées et définies par l'utilisateur. (Cattell, 2011)	Gère une variété de types de données complexes, y compris le multimédia, des objets hiérarchiques, des structures définies par l'utilisateur et des relations imbriquées. (Staab et Studer, 2009)
Gestion de type de données complexes	+	+	+	++	++	++

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Maturité du marché / Maintenance	Très matures, utilisés depuis des décennies. (Blatt, 1987)	Existe depuis longtemps mais reste moins populaire que le modèle relationnel. (Blatt, 1987)	Large adoption et outils de support matures	Manque encore d'expérience et de modèle de données universel. (Ijmter, 2014)	Continue d'émerger mais encore relativement immature. (Cattell, 2011)	Mature dans certains domaines mais leur adoption n'est pas très répandue. (Staab et Studer, 2009)
Maturité du marché / Maintenance	++	+	++	+	+	+
Extensibilité	Peu extensible en raison de sa rigidité structurelle. (Blatt, 1987)	Très extensible, permet de facilement ajouter de nouveaux types de relations de données. (Blatt, 1987)	Limité aux types de données et opérations prédéfinis. (Suri et Sharma, 2011)	Facilement extensible avec des types de données définis par l'utilisateur et l'héritage et le polymorphisme. (Suri et Sharma, 2011), (Ijmter, 2014)	Très extensible, permettant l'ajout de nouveaux champs sans mise à jour du schéma (Cattell, 2011)	Très extensibles grâce à leur capacité à ajouter de nouveaux concepts et relations sans affecter la structure existante. (Sir et al., 2015), (Staab et Studer, 2009), (Gruber, 1993)
Extensibilité	-	+	+	+	++	++

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Flexibilité de développement	Très grande rigidité par sa structure arborescente, peu flexible pour les relations complexes. (Blatt, 1987)	Flexible, s'adapte bien aux relations complexes et aux changements fréquents. (Blatt, 1987)	Moins flexible en raison de structures de tables rigides et la nécessité de faire correspondre les objets aux tables (impédance mismatch). (Suri et Sharma, 2011)	Très flexible car il s'intègre directement avec les langages de programmation orientés objet avec les technologies de bases de données. (Suri et Sharma, 2011), (Ijmter, 2014)	Très flexible avec une conception sans schéma. (Cattell, 2011)	Très flexible, utilise des sémantiques formelles pour définir des concepts et des relations. Cela permet des modèles riches et dynamiques. (Sir et al., 2015)
Flexibilité de développement	- -	+	+	++	++	++

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Interopérabilité	Limitée aux systèmes hiérarchiques compatibles. (Blatt, 1987)	Adaptables à différents systèmes. (Blatt, 1987)	Limitée, principalement utilisé dans des systèmes homogènes	S'intègre bien avec les langages de programmation orientés objet. Réduction de l'incompatibilité d'impédance entre les applications et les bases de données. (Suri et Sharma, 2011), (Ijmter, 2014)	Assez interopérable, pour plusieurs langages de programmation et technologies modernes mais manque de caractéristiques d'interopérabilité solide, comme d'un langage de requête standardisé comme le SQL. (Cattell, 2011)	Les ontologies facilitent l'interopérabilité entre différents systèmes et technologies via des standards ouverts comme OWL et RDF, permettant le partage et l'échange de données entre systèmes (Sir et al., 2015), (Gruber, 1993)
Interopérabilité	- -	+	+	+	+	++

Suite du tableau

Critères	Hiérarchique	En réseau	Relationnel	Orienté objet	NoSQL	Ontologie
Expressivité	Peu expressive, adaptée pour des relations simples. (Blatt, 1987)	Permet de modéliser des relations complexes. (Blatt, 1987)	Moins expressif, axé sur les relations prédéfinies et les requêtes structurées. (Suri et Sharma, 2011)	Permet une expressivité riche pour représenter des entités réelles avec des données et des relations complexes. (Suri et Sharma, 2011), (Ijmter, 2014)	Moins expressif que les bases de données SQL pour des requêtes complexes, mais offre une grande flexibilité dans la gestion de données variées et à grande échelle. (Cattell, 2011)	Grande expressivité avec prise en charge du raisonnement logique et de l'inférence, permet une représentation implicite des connaissances. (Staab et Studer, 2009), (Gruber, 1993)
Expressivité	- -	+	+	++	+	++
Total	0	13	14	16	12	17

ANNEXE B TABLEAU DESCRIPTIF DES *CLASSES* DE L'ONTOLOGIE

TABLEAU B.1 Descriptif des classes de l'ontologie Protege

Classes	**Définition**	**Champs associés**
Area	Zone fonctionnelle regroupant des ressources ou opérations selon leur emplacement ou finalité.	Area_ID
InspectionArea	Zone dédiée au contrôle qualité des produits selon des critères définis.	Area_ID
ProductionArea	Zone dédiée aux opérations de production	Area_ID
StorageArea	Zone dédiée au stockage de matières premières, composants ou produits finis.	Area_ID
BOM	*[L]iste de tous les matériaux nécessaires à la fabrication d'un produit, indiquant la quantité de chaque nécessaire.*	BOM_ID, BOM_Category
Calendar	Répertoire des périodes de disponibilité, incluant horaires de production et jours fériés.	Calendar_ID
Cost Center	Unité organisationnelle utilisée pour regrouper les coûts liés à l'exécution d'une opération	CostCenter_ID, CostCenter_Type
Count	Valeur représentant une quantité discrète ou un nombre d'occurrences mesurées ou attendues dans un processus de production.	Count_Scale, Count_Value

Suite du tableau

Classes	Définition	Champs associés
CountEquipmentInfees	Équipement utilisé pour enregistrer ou mesurer la quantité de matière première entrant dans une opération.	Count_Scale, Count_Value
CountEquipmentOutfees	Équipement utilisé pour enregistrer ou mesurer la quantité de produit ou matière sortant d'une opération.	Count_Scale, Count_Value
CountEquipmentReject	Équipement utilisé pour identifier ou mesurer les éléments rejetés durant la production.	Count_Scale, Count_Value
CustomProperty	Temps théorique ou mesuré pour une réalisation à spécifier	CustomProperty_Value, CustomProperty_ValueType
Duration	Temps théorique ou mesuré pour une réalisation à spécifier	Duration_Value
Delivery Duration	Durée de livraison	Duration_Value
Maximum Wait Duration	Durée maximale pendant laquelle une unité peut attendre entre deux opérations	Duration_Value
Minimum Move Duration	Durée minimale requise pour déplacer une unité entre deux postes de travail ou opérations.	Duration_Value
Minimum Queue Duration	Durée minimale pendant laquelle un produit peut attendre dans une file avant traitement	Duration_Value
Minimum Wait Duration	Durée minimale pendant laquelle une unité peut attendre avant l'execution d'une opération	Duration_Value
Standard Move Duration	Durée standard estimée pour le déplacement d'une unité entre deux étapes	Duration_Value

Suite du tableau

Classes	Définition	Champs associés
Standard Queue Duration	Durée d'attente moyenne considérée comme nominale pendant laquelle un produit peut attendre dans une file avant traitement	Duration_Value
External Processing	Étape de fabrication réalisée en dehors du site de production, généralement prise en charge par un prestataire externe.	ExternalProcessing_ID
Subcontracting	Forme de traitement externe dans laquelle une partie de la production est confiée à un sous-traitant, les composants nécessaires étant fournis par l'entreprise donneuse d'ordre.	Subcontracting_ID
Inspection Specification	Critères ou exigences qualité associés à une opération ou à un produit	InspectionSpecification_ID, InspectionSpecification_DecimalPlaces, InspectionSpecification_LowerLimit, InspectionSpecification_UpperLimit, InspectionSpecification_ResultCalculation, InspectionSpecification_TargetValue, InspectionSpecification_Tolerance
Plant	Entité physique ou logique représentant un site de production	Plant_ID

Suite du tableau

Classes	Définition	Champs associés
Process Segment	Ensemble des capacités nécessaires pour un segment de production	ProcessSegment_ID, ProcessSegment_BusinessProcess, ProcessSegment_CostingRelevency-Type, ProcessSegment_DenominatorforconvertingmasterrecipeandoperationUoM ProcessSegment_NumeratorforconvertingroutingandoperationUoM, ProcessSegment_ReferenceQuantity
Process Segment Component	Élément (matière ou sous-assemblage) requis en entrée d'un segment de processus pour en permettre l'exécution.	ProcessSegmentComponent_ID, ProcessSegmentCom-ponent_BOMNodeNumber, ProcessSegmentComponent_Marked-ForBackflush, ProcessSegmentComponent_Type
Process Segment Dependency	Relation de dépendance entre deux segments de processus, définissant l'ordre ou les conditions de leur exécution.	ProcessSegmentDependency_ID, *ProcessSegmentDependency_Timing-Factor*
Process Segment Dependency Type	Catégorie précisant la nature de la dépendance entre deux segments de processus (ex. : séquentielle, simultanée).	ProcessSegmentDependencyType_ID
Product	Résultat final ou intermédiaire d'un processus de fabrication, destiné à être stocké, livré ou intégré dans un autre produit.	Product_ID

Suite du tableau

Classes	Définition	Champs associés
Product Production Rule	Ensemble des informations *utilisées pour indiquer à une opération de fabrication comment produire un produit*	*ProductProductionRule_ID*
PRTControlProfile	Profil de configuration associé à un outil ou une ressource de production, définissant son comportement ou ses paramètres d'utilisation.	PRTControlProfile_ID
PurchaseContract	Accord contractuel définissant les termes et conditions d'achat de biens ou de services nécessaires à la production.	PurchaseContract_ID, PurchaseContract_Item, PurchaseContract_NetPrice
PurchasingInfoRecord	Informations d'achat associées à un fournisseur pour un matériel donné.	PurchasingInfoRecord_ID, PurchasingInfoRecord_CostElement
QATestSpecification	Ensemble de critères, méthodes et seuils utilisés pour vérifier la qualité d'un produit ou d'une opération.	QATestSpecification_ID, QATestSpecification_LotType
Quantity	Valeur numérique exprimant une grandeur mesurable utilisée dans un processus de fabrication (ex. : quantité de matière, de cycles, etc.).	Quantity_Value
Personal Quantity	Quantité de personnel nécessaire pour exécuter une opération donnée	Quantity_Value
Standard Work Quantity	Quantité standard de travail ou de ressources requise pour accomplir une opération dans des conditions normales.	Quantity_Value
Total Quantity	Quantité globale de ressources ou produits, incluant toutes les composantes ou étapes d'un processus.	Quantity_Value, TotalQuantity _CalculFormula

Suite du tableau

Classes	Définition	Champs associés
Usage Quantity	Quantité consommée ou utilisée d'un composant ou d'une ressource pour la réalisation d'une opération.	Quantity_Value, UsageQuantity_CalculFormula
Resource	Ensemble des moyens nécessaires à l'exécution d'une opération. Cela inclut les Production Resources/Tools (PRT) et les spécifications du personnel	-
Personal Specification	Ensemble des caractéristiques associées à une ressource humaine.	Person_ID, PersonalSpecification_WageGroup, PersonalSpecification_WageType
PRT	Ensemble des ressources non humaine.	PRT$_{I}temNo$
PRT_Material	Matériau requis au cours d'une opération, non inclus dans la BOM.	PRTMaterial_ID
PRT_Equipment	Équipement utilisé pendant une opération.	PRTEquipment_ID
PRT_Document	Document de référence utilisé lors d'une opération.	PRTDocument_ID
DocumentPart	Élément constitutif d'un document de production, permettant de référencer de manière plus précise ses sections.	DocumentPart_ID
PRT_Miscellaneous	Ressource temporaire ne relevant d'aucune catégorie prédéfinie.	PRTMiscellaneous_ID
ResourceType	Catégorie de classification des ressources facilitant leur classification et leur gestion.	Resource_Type
DocumentType	Catégorie de classement des documents selon leur contenu ou usage.	Resource_Type

Suite du tableau

Classes	Définition	Champs associés
EquipmentType	Classification des équipements selon leur fonction ou domaine d'usage.	Resource_Type
MaterialType	Catégorie regroupant les matériaux selon leur nature ou leur rôle en production.	Resource_Type
MiscellaneousType	Catégorie de ressources diverses ne relevant d'aucun autre type défini.	Resource_Type
Personal Class	Classification ou groupe de personnel partageant des caractéristiques communes en lien avec la production.	Resource_Type
Sequence	Ordre ou arrangement des opérations dans une gamme de fabrication	Sequence_ID
Sequence Category	Type d'organisation des séquences dans une gamme (par ex. : série principale, alternative, parallèle).	SequenceCategory_ID
Status	État courant d'un objet dans le système (ex. : actif, obsolète, en cours, terminé).	Status_Value
Supplier	Entité externe fournissant des composants, matières premières ou services nécessaires à la production.	Supplier_AccountNo, Supplier_Currency, Supplier_ID
Unit Of Measure	Unité utilisée pour exprimer une grandeur physique dans le système (ex. : kg, m, h, pièce).	Unit$_I D, Unit_S ymbol$
User	Acteur humain ou profil utilisateur interagissant avec le système pour effectuer des opérations ou consulter des données.	User_ID
Version	Itération particulière d'un élément de la Product Production Rule	ValidityStartDate, ValidityEndDate

Suite du tableau

Classes	Définition	Champs associés
WorkCenter	Ensemble d'équipements ou de postes de travail regroupés pour l'exécution d'un ensemble d'opérations spécifiques.	WorkCenter_ID, WorkCenter_Type

ANNEXE C TABLEAU DESCRIPTIF DES *OBJECT PROPERTIES* DE L'ONTOLOGIE

TABLEAU C.1 Descriptif des Object Properties

Object Properties	**Domain**	**Range**	**Cardinalité**
changed By	Product Production Rule	User	some
created By	Product Production Rule	User	exactly 1
has A Delivery Duration	Purchase Contract	Delivery Duration	
has Branch Process Segment	Sequence	Process Segment	some
has Control Profile	PRT	Control Profile	exactly 1
has Cost Center	Standard Work Quantity	Cost Center	exactly 1
has Counting Equipment	Count	Equipment	max 1
has Dependency	Process Segment	Process Segment Dependency	
has Dependency Type	Process Segment Dependency	Process Dependency Type	exactly 1
has For Duration	Process Segment	Duration	min 1
has For Standard Work Quantity	Process Segment, PRT	Standard Work Quantity	min 1
has For Unit	Count, Standard Work Quantity, Duration, Process Segment Dependency, Usage Quantity, Process Segment, Product Production Rule, Inspection Specification, Personal Specification	Unit of Measure	exactly 1

Suite du tableau

ObjectProperties	Domain	Range	Cardinalité
has For Usage Quantity	PRT	Usage Quantity	exactly 1
has Total Quantity	PRT	TotalQuantity	exactly 1
has TrackRecord	Product Production Rule, Process Segment, Process Segment Dependency, Resource, Inspection Specification, Product, Sequence, Process Segment Component, BOM, Cost Center, Plant, PRT Control Profile, Purchase Contract, QA Test Specification, Workcenter	Track Record	exactly 1
has Version	Sequence, Product, Process Segment Component, Process Segment, Resource, Product Production Rule, Inspection Specification		
is In Plant	Work Center	Plant	exactly 1
To Produce	Product Production Rule	Product	exactly 1
uses Component	Process Segment	Process Segment Component	min 1
uses Resource	Process Segment	Resource	min 1
uses Work Center	Process Segment	Work Center	exactly 1
hasBOM	Process Segment	BOM	exactly 1
hasBOMVariant	Process Segment	BOM	some
hasQATestSpecification	Process Segment	QATestSpecification	some
hasInspectionSpecification	QATestSpecification	InspectionSpecification	some