



HAL
open science

Modélisation des systèmes ferroviaires -objectifs, approches et problématiques

David Camarazo, Sana Debbech, Ana Roxin, Annabelle Gillet

► **To cite this version:**

David Camarazo, Sana Debbech, Ana Roxin, Annabelle Gillet. Modélisation des systèmes ferroviaires -objectifs, approches et problématiques. eduBIM 2023 - 9e édition des Journées de l'enseignement et de la recherche autour du BIM et de la maquette numérique, Université de Bourgogne, Nov 2023, Dijon, France. hal-04353184

HAL Id: hal-04353184

<https://hal.science/hal-04353184>

Submitted on 19 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modélisation des systèmes ferroviaires – objectifs, approches et problématiques

David CAMARAZO^{1,2}, Sana DEBBECH², Ana ROXIN¹,
Annabelle GILLET¹

¹ LIB (Laboratoire d'Informatique de Bourgogne) EA7534, Université de Bourgogne,
21000 Dijon

²IRT Railenium, Technopole Transalley, 59803 Valenciennes,
{david.camarazo, sana.debbech}@railenium.eu
{ana-maria.roxin, annabelle.gillet}@u-bourgogne.fr

Résumé : Les systèmes ferroviaires se caractérisent par des besoins critiques en termes de sûreté de fonctionnement, de contraintes métier, technologiques et opérationnelles liées à l'environnement du système, qu'il est nécessaire d'intégrer dès la phase de conception. La modélisation de tels systèmes repose sur diverses pratiques et acteurs, soulevant des problèmes d'ambiguïté sémantique et induisant des erreurs de conception. L'ingénierie dirigée par les modèles est souvent employée afin de garantir la conformité d'un prototype par rapport à un ensemble d'exigences. Cependant, les ingénieurs peuvent rencontrer des difficultés quant à l'élaboration et la compréhension des modèles sous-jacents. Cet article étudie les causes et les conséquences de ces problèmes de modélisation des systèmes ferroviaires, et expose un tour d'horizon des solutions qui ont été relevées dans la littérature. Nous terminons avec la présentation d'une approche prospective pour la vérification des différentes phases du processus de modélisation ferroviaire.

Mots-clés : Système ferroviaire, modèles conceptuels, Ingénierie dirigée par les modèles, UML, Capella, Web sémantique

Abstract : In the railway domain, we must develop railway systems respecting user's needs. Such systems are complex, implying humans and multiple technical objects, and said to be critic. This means that a failure of the system cause serious consequences on user's lives. In this context, we use model-based system engineering to guarantee the consistency of the prototype regarding the security constraints and the user's needs. However, engineers may encounter problems regarding the modelling and understanding of the models. This paper presents causes and consequences of such modelling problems in railway domain and introduce some solutions found in the literacy.

Keywords : Railway system, conceptual models, Model Based System Engineering, UML, Capella, semantic Web

1 Introduction

Le domaine ferroviaire est un domaine complexe dû à la taille, en nombre d'éléments, de l'infrastructure nécessaire au fonctionnement du matériel roulant.

Un système ferroviaire est composé de (Walter Schön, 2013) :

- Dispositifs techniques (matériel roulant, infrastructures, composants matériels par exemple)
- Interventions humaines (agents de voie, agents de signalisation, etc.)
- Environnement dans lequel opère le système (tout ce qui est présent/existe avant que le système ne soit conçu)

Dans un système ferroviaire, les composants de l'infrastructure sont utilisés par le matériel roulant. Le terme « matériel roulant » désigne n'importe quel type de train (train léger, train de fret, train télé-conduit par exemple). L'infrastructure comprend l'ensemble des dispositifs nécessaires au fonctionnement du matériel roulant. Cela intègre les voies qui offrent au matériel roulant un support pour rouler, mais aussi les appareils de télécommunications, d'alimentation électrique et enfin la signalisation. Chacun de ces composants est lui-même un élément complexe, c'est-à-dire une agrégation de plusieurs éléments (l'énergie par exemple peut comprendre le système de distribution électrique, le système de commande ou la source d'énergie). C'est pourquoi les systèmes ferroviaires sont des systèmes de systèmes (Bunge, 1979). Ces systèmes sont complexes à modéliser et dits critiques en implémentation. Un système critique de sécurité (SCS) (Debbech S. a.-D., 2018) est un système dont une défaillance entraîne des conséquences graves sur les utilisateurs (leurs vies peuvent être mises en danger). Ce besoin critique en termes de sûreté de fonctionnement doit être pris en compte dès la phase de conception (Debbech S, 2018). Dans ce contexte, l'Institut de Recherche Technologique (IRT) Railenium et ses partenaires de la filière ferroviaire ont choisi un processus de modélisation basé sur les modèles, selon les recommandations de la norme NF EN 50126-2:2018 « Applications ferroviaires - Spécification et démonstration de la fiabilité, de la disponibilité, de la maintenabilité et de la sécurité (FDMS) - Partie 2 : approche systématique pour la sécurité » (50126-2:2018, 2018). Cette phase de modélisation est cruciale puisqu'elle produit les modèles qui vont servir pour la conception du prototype. La qualité du prototype dépend donc de la qualité de la phase de modélisation.

Les pratiques de modélisation de ces systèmes sont guidées principalement par le retour d'expérience (REX), les normes d'ingénierie système (IEEE, ISO/IEC/IEEE International Standard - Systems and Software engineering Methods and tools for model-based systems and software engineering, 2023) et les normes du domaine d'application fournies par le CENELEC (CENELEC, 2023). L'ingénierie dirigée par les modèles (IDM) est employée par les acteurs du domaine afin de garantir la conformité du prototype aux besoins de l'utilisateur et aux exigences d'un système critique (Debbech S. a., 2020)

Cet article présente les réflexions menées jusqu'ici dans le cadre d'une thèse de doctorat financée par l'IRT Railenium. L'objectif est de mettre en évidence les problématiques métier comme scientifiques associées aux pratiques de modélisation des systèmes ferroviaires. Nous commençons par présenter, dans la section 2 les notions d'ingénierie des modèles et d'ingénierie des connaissances utilisées dans cet article. La section 3 présente le processus de modélisation de systèmes ferroviaires et les limites de ce processus. La section 4 relève certaines approches qui ont déjà été proposées pour traiter la vérification de modèles. La section 5 présente notre approche et la section 6 conclut cet article.

2 Fondements scientifiques

2.1 Ingénierie dirigée par les modèles

L'ingénierie dirigée par les modèles est une méthode générique d'ingénierie dans laquelle les modèles sont au centre du processus de conception. Le système déployé est issu d'une suite de transformation de modèles (Truyen, 2006).

Afin de mieux comprendre ce qu'est l'IDM et la façon dont elle est implémentée en pratique, les paragraphes suivants définissent ses concepts clés.

Un **modèle** est une simplification d'un système construite autour d'un but (Favre, 2004). Le modèle doit pouvoir répondre à des questions concernant le système modélisé à la place du système. Certains articles comme (Kleppe, 2003) restreignent cette définition, en ajoutant qu'un modèle doit être exprimé dans un langage avec une syntaxe formelle qui peut être traité par la machine. Un modèle représente un système.

Plusieurs définitions existent pour le concept de « **langage** ». On considère ici la définition traduite du glossaire de la terminologie de l'ingénierie logicielle de l'IEEE : « Un moyen de communication avec de la syntaxe et de la sémantique consistant en un ensemble de

représentations, de conventions et de règles associées utilisées pour véhiculer de l'information » (IEEE:610.12-1990).

Méta-modèle : Un méta-modèle (Seidewitz, 2003) est un modèle qui spécifie et décrit les modèles valides du langage de modélisation.

Comme expliqué dans (Klaus Pohl, 2012) on peut découper le processus de modélisation en quatre vues qui répondent chacune à une question primordiale :

- **Vue des besoins** : Quels sont les besoins, fonctionnels et non-fonctionnels, que doit satisfaire le système ? Cette vue du système ne montre que les besoins des utilisateurs et les exigences auxquels doit se conformer le système. Elle est basée sur les besoins des utilisateurs, le contexte d'utilisation ainsi que sur des documents légaux. Cela permet une distinction nette entre le problème et sa solution qui permettra de réutiliser certains éléments de la solution si on retrouve des problèmes similaires dans d'autres projets.
- **Vue fonctionnelle** : Que doit faire le système pour satisfaire les besoins décrits précédemment ? Cette vue du système donne les fonctions proposées par le système pour satisfaire les besoins des utilisateurs en restant conforme aux exigences du système. Cette vue est nécessaire pour faire le lien entre les besoins et la solution proposée.
- **Vue logique** : Comment le système doit se comporter pour satisfaire les besoins ? Cette vue du système donne une décomposition des fonctions en architecture de composants logiques. On décrit ici une solution logique, indépendante de toute technologie qui sera utilisée lors du déploiement. Cette solution logique pourra donc être réutilisée pour être redéployée avec des technologies différentes.
- **Vue technique** : Comment le système doit se comporter concrètement, pour satisfaire les besoins (vue technique) ? Cette vue du système présente les informations techniques nécessaires au déploiement de la solution. On fait ici le lien entre les composants logiques et la solution concrète déployée.

L'IDM répartit ces vues dans trois niveaux de modélisation (Truyen, 2006) :

- Le modèle indépendant du calcul (CIM pour Computational Independent Model qui modélise la vue des besoins
- Le modèle indépendant de la plateforme (PIM pour Platform Independent Model) qui modélise les vues fonctionnelle et logique - dans le processus de modélisation ferroviaire, les modèles UML sont utilisés pour représenter les modèles conceptuels correspondants au PIM (Rahmig, 2022).
- Le modèle spécifique à la plateforme (PSM pour Platform Specific Model qui modélise la vue technique - le niveau PSM correspond à la modélisation dans un langage spécifique.

Le langage UML (Unified Modeling Language) est un langage de modélisation semi formel. En se référant à la documentation (OMG, 2003) de l'Object Management Group (OMG) UML est défini comme «Un langage pour spécifier, visualiser, construire, et documenter les entités des systèmes logiciels, ou des business-model et autres systèmes non logiciels». Les objets et les instances sont représentés par des rectangles dans lesquelles sont affichés leurs attributs. Les relations entre ces objets sont représentées par des symboles graphiques e.g. flèches, arcs, lignes etc. Les modèles UML sont construits selon des méta-modèles, eux même construits suivant le méta-méta-modèle IDM, à savoir le Meta Object Facility (MOF) (OMG, 2019). Les notions de modèle et de méta-modèle sont relatives. Le méta-modèle d'un modèle peut très bien être un modèle d'un autre méta-modèle.

2.2 Ingénierie dirigée par les connaissances

Une ontologie est une « spécification explicite et formelle d'une conceptualisation partagée d'un domaine de connaissances » (Studer, 1998). Selon (Guarino, 1995), une conceptualisation est une structure sémantique qui encode en intension les règles implicites définissant la structure d'une portion de la réalité. Une ontologie est dite explicite car les types de concepts, les relations

qu'ils partagent et les contraintes qui s'appliquent sur eux sont explicités. Les connaissances explicitées par l'ontologie sont partagées par les acteurs qui doivent l'utiliser. Les connaissances représentées par l'ontologie sont issues d'un consensus et donnent une vision commune aux acteurs. Une ontologie est formelle car les informations qu'elle encode peuvent être traitées par un ordinateur. Enfin une ontologie ne peut pas représenter l'ensemble des connaissances humaines et se limite à un domaine d'activité.

Les ontologies sont construites en utilisant les langages de la logique de description (Description Logic DL). La logique de description permet de n'utiliser que les constructeurs nécessaires pour avoir des ontologies décidables sur lesquelles on peut appliquer des algorithmes appelés raisonneurs qui classent les connaissances en un temps fini) et efficaces en implémentation (avec un temps d'exécution des requêtes fini). Nous introduisons ici les connaissances clés de la logique de description en nous basant sur (Britz, 2020).

La logique de description est une famille de langages logiques utilisée pour représenter des connaissances.

Une théorie en logique de description emploie trois ensembles pour décrire une ontologie : l'ensemble C des concepts l'ensemble R des relations et l'ensemble I des instances ou individus.

Une ontologie comprend un schéma terminologique (ou TBox) ainsi qu'un ensemble d'assertion (ou ABox). La T-box contient les connaissances terminologiques (hiérarchie de classe) ainsi que les contraintes sur les éléments de C et R. La A-box spécifie les éléments de l'ensemble I (ou les instances) en les reliant aux concepts de C et aux relations de R.

Par rapport aux approches à base de modèles, les approches à base d'ontologies ont l'avantage de pouvoir exploiter des algorithmes pour inférer des connaissances explicites à partir des connaissances représentées par la T-box et la A-box. Par exemple si on sait que tout humain a un parent et que « Pierre » est un humain, l'ontologie peut inférer que Pierre doit avoir un parent.

Une ontologie peut être écrite dans plusieurs langages informatiques (OWL, RDFS) de sorte à pouvoir être traitée par un ordinateur. Resource Description Framework Schema (RDFS) est un langage de description d'ontologies qui permet de spécifier une hiérarchie entre des concepts et une hiérarchie entre des propriétés. Cependant il ne permet pas d'exprimer des contraintes sur les concepts. Par exemple on ne peut pas écrire que le concept train et passager sont des concepts disjoints (aucune instance de passager n'est une instance de train et vice-versa). OWL est aussi un langage de description d'ontologies qui apporte les éléments nécessaires pour répondre aux limites de RDFS. Par rapport à l'exemple précédent, OWL contient le prédicat owl:disjointWith qui permet de spécifier que 2 classes sont disjointes. OWL permet de faire la distinction entre propriétés objet (propriétés qui relie des concepts, exemple : « estSupérieurHiérarchiqueDe ») et propriété de données (propriété qui affecte une valeur à un objet, exemple : « estAgéDe »). OWL permet aussi de définir des propriétés d'objet inverses notamment via le prédicat owl:inverseOf.

3 Modélisation de systèmes ferroviaires – approche et problématiques

3.1 Approche métier pour la modélisation des systèmes ferroviaires

L'objectif actuel du processus de conception de systèmes ferroviaire adopté par l'IRT Railenium et ses partenaires est de déployer une solution conforme aux exigences du système (Chouchani, 2022). Pour atteindre cet objectif de conformité entre les exigences et le prototype, le problème et la solution, l'IRT et ses partenaires appliquent les principes de l'IDM. Des ingénieurs de différentes spécialités (ingénieur des exigences, ingénieur en génie électrique, etc.) ainsi que des constructeurs de matériel roulant et des gestionnaires d'infrastructure procèdent à une phase de modélisation avant de lancer la phase de déploiement du prototype. Cela a pour but d'amener une transition progressive et cohérente des besoins abstraits vers le déploiement d'une solution concrète. La cohérence du processus de modélisation est ce qui caractérise la correspondance sémantique des modèles produits tout au long de la phase de modélisation.

La modélisation de systèmes ferroviaires est un processus itératif et incrémental composé de 4 étapes comme illustré sur la Figure 1. Ces étapes sont décrites dans les paragraphes suivants.

La première phase du processus de modélisation est celle de l'« **analyse des besoins** ». Son objectif est de poser les besoins auxquels le système doit répondre. On se place ici du point de vue utilisateur et on liste les besoins de l'utilisateur que le système doit satisfaire. Cette phase primordiale va servir de base pour la conception du système. L'idée est que la conception du système doit être orientée par le besoin de l'utilisateur qui fait office d'objectif à atteindre pour le système. Cette phase est menée côté utilisateur par différents acteurs pouvant comprendre un constructeur du matériel roulant ou un gérant de l'infrastructure. À l'issue de cette phase, on obtient une expression fonctionnelle des besoins (EFB). Il s'agit d'un tableau listant les besoins exprimés sous forme de fonction.

La seconde phase est celle de la « **Conceptualisation** » et vise à modéliser ce que doit faire le système, sans rentrer dans les détails sur comment il le fait. On se place cette fois-ci du point de vue du système pour répondre à la question « Que doit faire le système pour satisfaire les besoins de l'utilisateur ». On commence aussi à réfléchir pour la phase suivante qui adressera la question « Comment le système répond aux besoins ». On reste ici sur une approche logique, indépendante des technologies utilisées. Cette phase doit encore être assez abstraite pour

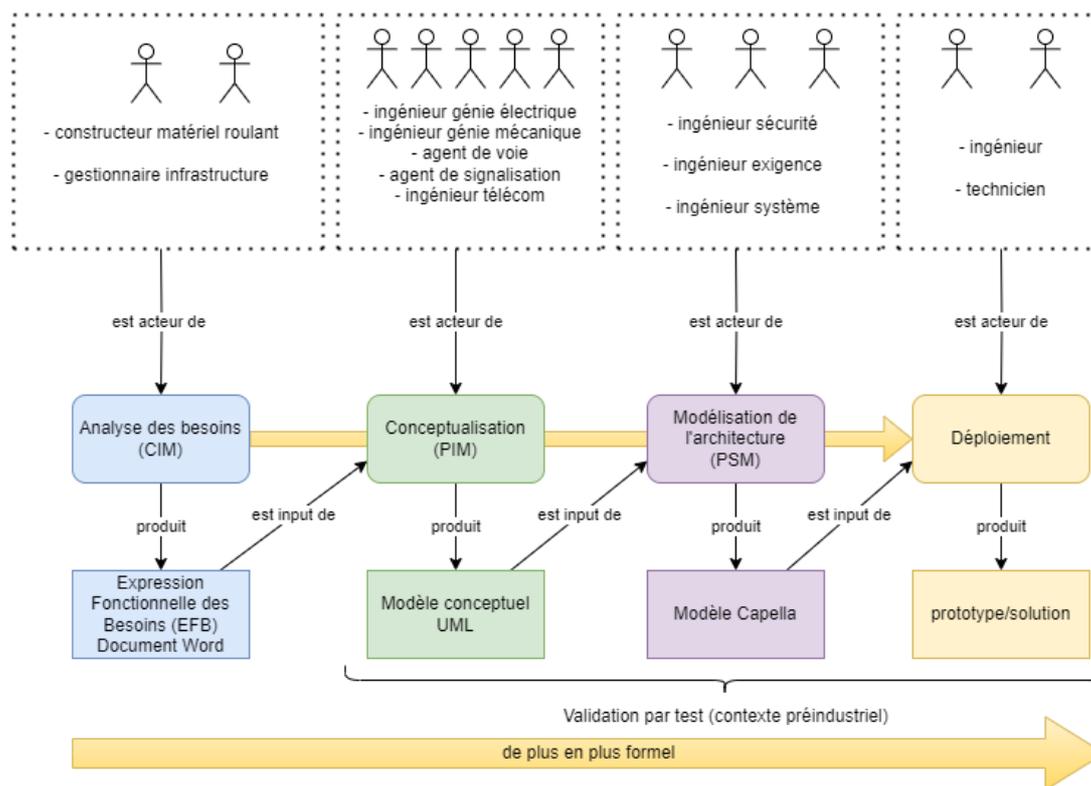


Figure 1. Illustration du processus de modélisation d'un système ferroviaire tel qu'utilisé à l'IRT

faciliter la réutilisation des solutions trouvées ainsi que garantir une certaine modularité dans le système. Cette phase se situe donc à un niveau fonctionnel. Elle fait intervenir des ingénieurs de différentes spécialités (génie électrique, génie mécanique ou télécom) ainsi que des agents de voie ou de signalisation. Les connaissances de ces nombreux acteurs permettent d'avoir une vue organisée et complète du système et de ses différents composants. Par exemple on peut modéliser ainsi toute l'infrastructure qui comprend la signalisation, la voie, l'énergie et la télécommunication. À l'issue de cette phase on obtient un modèle conceptuel UML.

La troisième phase du processus de modélisation, « **Modélisation de l'architecture** » a pour but de produire un modèle permettant de répondre aux questions concrètes de déploiement.

Cette phase se positionne toujours du point de vue du système et permet de répondre à la question « Comment doit se comporter le système concrètement pour satisfaire les besoins de l'utilisateur ». Contrairement à la phase précédente qui adoptait un point de vue fonctionnel, ici on adopte un point de vue technique en tenant compte des contraintes liées au déploiement des fonctionnalités dans la réalité. Cette phase fait intervenir des ingénieurs de sécurité, des exigences ou des ingénieurs système par exemple. À l'issue de cette phase on obtient un modèle d'architecture en Capella DSML (Voirin, 2016).

Enfin, la quatrième et dernière phase est celle du « **Déploiement** ». Menée par des ingénieurs et des techniciens de différentes spécialités, cette phase ne produit pas de modèle contrairement aux précédentes mais s'appuie sur celui produit lors de la phase de modélisation pour produire un prototype. Même si cette phase ne produit pas de modèle, elle fait partie intégrante du processus de modélisation dans la mesure où elle permet de valider et corriger si besoins, les modèles produits aux phases précédentes.

3.2 Analyse du processus de modélisation ferroviaire

Ce processus de modélisation est conforme aux principes promus par l'IDM. L'analyse des besoins répond exactement aux questions auxquelles doit répondre le CIM, en ne listant que les besoins d'un point de vue utilisateur et en faisant abstraction de toute information sur la manière dont les besoins sont satisfaits. La phase de conceptualisation permet de produire des modèles qui montrent d'un point de vue logique comment les besoins des utilisateurs seront satisfaits, sans détailler comment ces solutions seront déployées dans la réalité. En cela la conceptualisation produit exactement le PIM. Enfin la modélisation de l'architecture présente les informations propres au déploiement ou à l'implémentation et produit donc exactement le PSM.

Le processus de modélisation n'est pas si linéaire que la Figure 1 pourrait le faire penser. Par exemple, même si elle fait partie de la phase de conceptualisation, il est nécessaire de conduire une étape d'ingénierie des exigences après l'analyse des besoins pour faire le lien entre les besoins, très abstraits et qui adoptent le point de vue de l'utilisateur, et la conceptualisation qui adopte le point de vue du système. Il est important de distinguer les deux notions, besoins et exigences, car elles ne répondent pas à la même question. Les besoins permettent de répondre à la question « À quel besoin de l'utilisateur répond le système » tandis que les exigences (Davis, 2013) permettent de répondre à la question « Quels sont les conditions/critères que doit vérifier le système pour satisfaire les besoins de l'utilisateur ». L'ingénierie des exigences permet donc d'entamer la réflexion sur les fonctionnalités que doit proposer le système pour satisfaire les besoins abstraits de l'utilisateur (domaine du problème) mais aussi les exigences plus concrètes du système (domaine de la solution). La Figure 2 illustre la transition entre la phase d'analyse des besoins et la phase de conceptualisation.

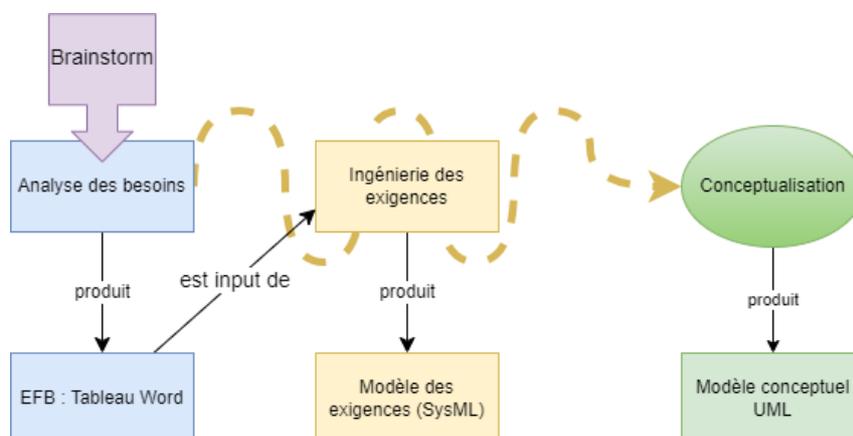


Figure 2. Zoom sur le passage des besoins aux modèles conceptuels

Il est également important de préciser, lors de la phase de modélisation de l'architecture, les ingénieurs ne consultent pas uniquement les modèles produits lors de la phase de conceptualisation. Ils consultent certes le modèle conceptuel et le modèle des exigences (souvent un diagramme Systems Modeling Language (SysML)) mais également l'expression fonctionnelle des besoins produite lors de la phase d'analyse des besoins.

Il est à noter que chaque phase de modélisation est complexe et nécessite plusieurs étapes. Prenons l'exemple de la modélisation de l'architecture. Il s'agit d'une phase comportant plusieurs étapes et suivant une méthodologie à part entière. Dans le domaine ferroviaire, c'est la méthodologie ARCADIA qui est utilisée. Cette méthode comporte 5 étapes : l'analyse opérationnelle, l'analyse des besoins du système, l'architecture logique, l'architecture physique et la description de la structure du produit fini. Cette méthode reprend les principes de l'IDM, on part d'un modèle des besoins pour aller vers un modèle décrivant l'architecture physique du système. On a donc un processus de modélisation à part entière pour réaliser une phase dans le processus de modélisation d'un système ferroviaire.

3.3 Problématiques métier

Dans la pratique, ce processus de modélisation ne se poursuit pas sans rencontrer de problèmes. Des spécifications peuvent changer et des erreurs de modélisations peuvent se produire. A travers les différents projets menés au sein de la filière ferroviaire, des difficultés dans la communication entre acteurs ont pu être observées. Les travaux menés dans la filière ferroviaire font état de problème d'échange de données, avec des formats qui ne correspondent pas. Les ingénieurs de recherches de IRT Railenium constatent des problèmes quant aux retours d'erreurs ou sur les modifications effectuées lors du processus de modélisation ferroviaire (Debbeck S. P.-D., 2019). Les modifications et les erreurs sont souvent rapportées par mail ou oralement entre certains acteurs, entraînant une documentation incohérente qui devient une source d'erreur. Les ingénieurs de recherche constatent également des abus de langage, ou des contre-sens utilisés dans les documents partagés entre les acteurs. La conséquence est un ajout important de réunion superflus pour expliquer ou réexpliquer le sens des termes (Chouchani N. a., 2023). Toutes ces erreurs entraînent des réunions supplémentaires, du travail supplémentaire pour les acteurs ainsi que du retard dans les projets. Tout ceci a un coût que les industries du secteur ferroviaire souhaiteraient baisser.

Le processus de modélisation repose sur des **acteurs humains pluridisciplinaires**. En résulte une collaboration difficile entre ces acteurs qui ont chacun un vocabulaire métier différent et une interprétation différente de mêmes termes métiers. De plus, chaque acteur utilise ses propres outils, le processus de modélisation fait donc intervenir plusieurs outils non interopérables. **La multiplicité des outils et langages de modélisations** ainsi que la pluridisciplinarité des acteurs entraînent des erreurs d'interprétation, de modélisation et in-fine, un prototype qui n'est pas conforme aux attentes des utilisateurs. Le processus de modélisation impacte la cohérence de l'information. Il est en effet difficile de garantir une conformité de la solution déployée par rapport aux besoins exprimés et aux exigences du domaine ferroviaire (notamment liées à la sécurité).

Pour illustrer ces problèmes de communication et de compatibilité, on se concentre sur la modélisation de l'architecture. Dans la pratique, l'utilisation de la méthode ARCADIA utilisée pour la modélisation de l'architecture repose sur l'usage d'un outil : Capella. Capella et ARCADIA ont été mis au point par Thalès qui a promu Capella auprès de la filière ferroviaire. Capella a donc été largement adopté par les acteurs du ferroviaire. Le problème aujourd'hui vient du fait que le langage Capella DSML ne suivant pas les standards du OMG (Object Management Group), il est difficile de comparer ou de lier les modèles Capella DSML aux modèles UML ou SysML. L'allocation des besoins aux composants, la notion de fonction et l'allocation de fonction aux instances sont des concepts importants en Capella DSML et absents du langage SysML. Les différences entre les langages de modélisation gênent l'interopérabilité des modèles Capella DSML et des modèles UML/SysML. Il est pourtant utile et nécessaire de comparer les modèles Capella aux modèles UML/SysML pour vérifier la cohérence des

modèles tout au long du processus de modélisation. Ces différences gênent également la compréhension et la communication entre acteurs qui n'ont pas tous la même formation et ne seront pas à l'aise avec les mêmes outils, certains préfèrent Capella, d'autres les modèles SysML.

3.4 Problématiques scientifiques

Par rapport aux problématiques métier précédemment identifiées, plusieurs problématiques scientifiques apparaissent.

La multiplicité des outils et la multidisciplinarité des acteurs entraînent des problèmes de communications important. Ces problèmes sont d'autant plus importants que les systèmes ferroviaires sont très complexes, et font intervenir de nombreux acteurs. Il est aujourd'hui difficile pour les parties prenantes d'un projet du ferroviaire d'avoir une vision précise et complète du système à modéliser et de l'avancement de la modélisation. Le nombre de modèles produits et leurs tailles en nombre d'entité à manipuler les rend difficile à traiter pour des humains. Les acteurs de la filière cherchent donc de nouveaux outils pour les assister dans le processus de modélisation. Cependant, simplement enregistrer de l'information ne fournit pas une assistance suffisante. Pour améliorer la communication et assister les décisions dans le processus de modélisation, il est nécessaire de pouvoir assurer une interprétation unique des informations échangées par les différents acteurs. Il y a donc un enjeu autour de l'aspect sémantique de l'information. De plus, dans le cas du processus de modélisation de systèmes ferroviaires, faisant intervenir différentes personnes venant de différentes organisations, il y a également un enjeu autour de la mise en commun et du partage de la sémantique des informations. Il y a donc un besoin d'interopérabilité sémantique dans le processus de modélisation de système ferroviaire.

Dans cette thèse, nous nous intéressons donc à l'interopérabilité sémantique. Plus particulièrement, nous nous intéressons à l'échange de modèles entre acteurs du ferroviaire, et au sens (sémantique) des modèles partagés. Les problématiques scientifiques associées sont détaillées dans la section 5.

4 Approches relevées dans la littérature pour la vérification automatisée de modèles

Cette section présente des approches qui ont déjà été testées ou déployées pour adresser le problème de la vérification des modèles. Nous présentons ici 3 approches : Les méthodes formelles, les annotations de modèles et la méthode proposée par Airbus (Hu X. a., 2022).

4.1 Méthodes de vérification formelles

Les méthodes formelles sont des méthodes qui visent à automatiser la vérification d'un système par le biais de méthodes mathématiques. Dans le ferroviaire, ces méthodes sont utilisées, par exemple, pour vérifier formellement qu'un système de contrôle des voies est conforme aux besoins du réseau.

L'idée est de modéliser le système à contrôler, les contraintes de sécurité ainsi que les opérations du système de contrôle dans un langage formel qui permet d'exprimer des théories logiques, et de vérifier que l'ensemble des formules est satisfiable ((Busard, 2015), (Kanso, 2009), (Haxthausen, 2011)). Il n'est pas question ici d'explicitier ou de sémantiser un modèle source, ni de les interroger.

Ces méthodes ne permettent pas de créer une terminologie commune et ne structure pas l'information à un niveau abstrait. Alors qu'elles sont aujourd'hui utilisées, les problèmes sont toujours présents dans le processus de modélisation ferroviaire. Ces méthodes ne suffisent pas à assister les acteurs dans leurs tâches de modélisation, ce qui s'explique par le fait qu'elles ne sont pas prévues pour établir une communication entre l'homme et la machine. Avec ces méthodes, il n'est pas question d'interroger, ni même de partager de l'information mais simplement de la traiter par calcul pour vérifier sa cohérence.

Certaines méthodes sont plus automatisées que d'autres, plus détaillées que d'autres, mais la principale différence entre les méthodes relevées est le langage utilisé pour exprimer formellement les modèles.

4.2 Annotation des modèles

Pour ajouter une sémantique compréhensible par l'homme et par la machine sur les modèles, et pour assurer que cette sémantique soit partagée entre les acteurs, plusieurs méthodes ont été proposées pour intégrer des ontologies au processus de modélisation via un système d'annotation. Cette approche postule qu'il est nécessaire d'explicitier les connaissances métier qui sont implicites dans les modèles comme expliqué dans (Hacid, 2016). Dans cette approche, L'ontologie sert à expliciter formellement les concepts implicitement manipulés par les modèles. Les annotations servent à faire le pont entre les classes des modèles et les concepts de l'ontologie. Pour illustrer cette approche, nous exposons ici une méthode détaillée dans (Liao, 2014). Nous avons choisi de présenter cette méthode car elle reste assez générique et ne formule pas d'hypothèse sur les modèles utilisés.

L'article montre que pour expliciter la sémantique des modèles il est nécessaire d'explicitier la sémantique du domaine mais également la sémantique des structures des modèles.

Les ontologies sont utilisées pour expliciter la sémantique du domaine et la sémantique de la structure des modèles.

Pour établir le système d'annotation, l'article définit le schéma d'annotation comme un quintuple composé de :

- E un ensemble des entités du modèle à relier à l'ontologie
- P un ensemble d'ensemble d'éléments appartenant aux ontologies de domaines qui décrivent les éléments de E (parfois, plusieurs éléments de P sont nécessaires pour décrire un élément de E, d'où P est un ensemble d'ensemble)
- SR qui donne la nature de la relation entre un élément de E est un ensemble d'élément de P : équivalence, plus général, plus spécifique, chevauchement, disjoint. Cette liste est exhaustive et SR est composé de cinq sous-ensembles, un pour chaque élément de la liste
- MME un ensemble d'élément de l'ontologie représentant le méta-modèle décrivant la structure sémantique de E
- MR l'ensemble des relations "is_instance_of" entre méta-modèles et modèles

Ce schéma d'annotation permet le raisonnement sur les modèles. L'article ne détaille pas qui annote les modèles, ni comment. La méthode introduit une entité, le SAPA (Semantic Annotation and Processing Agent), qui s'occupe de gérer les interactions entre les modèles, les annotations, les ontologies et le raisonneur. Un cas d'étude est présenté pour lequel un SAPA a été développé : SAP-KM. Le SAPA permet le raisonnement sur les modèles en établissant le lien entre les ontologies et les modèles via les annotations et en établissant le lien entre le

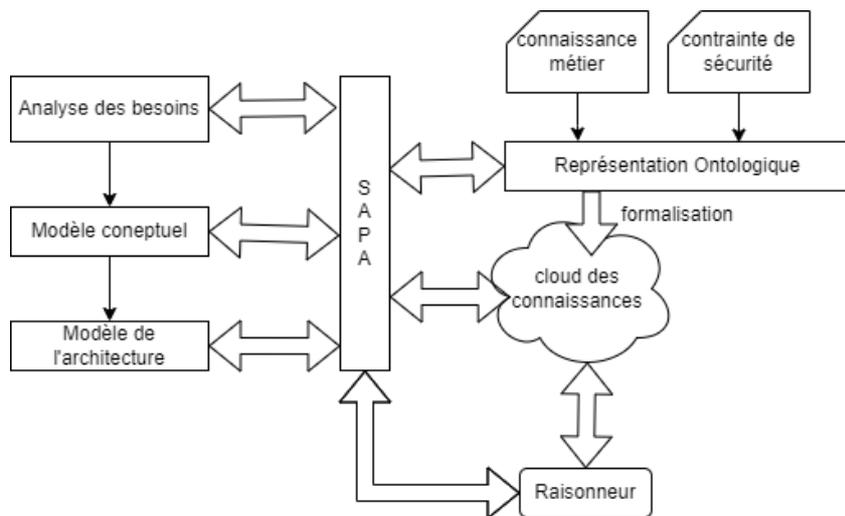


Figure 3. Communication entre les outils de la méthode par annotation présentée dans (Liao, 2014)

raisonneur et le modèle annoté (les liens entre les différents éléments de la méthode sont illustrés sur la Figure 3). Le raisonnement permet de valider et d'assister la conception des modèles en relevant les incohérences et en explicitant des inférences.

Un avantage de cette méthode est qu'elle ne fait pas d'hypothèse sur les outils ou sur les langages de modélisation. Elle laisse donc les acteurs responsables de la modélisation libre de choisir leurs outils et leurs langages de modélisation. Cependant cette méthode nécessite le développement d'outils et d'interfaces pour faire le lien entre les modèles, les annotations et les ontologies, comme l'illustre le cas d'usage présenté dans l'article (Liao, 2014).

4.3 Méthode pour les systèmes aéronautiques

Cette section détaille l'approche expérimentée par des chercheurs d'Airbus pour modéliser des systèmes aéronautiques. Les détails de l'approche sont exposés dans l'article (Hu X. a., 2022). Cette approche repose, comme la précédente, sur des ontologies pour sémantiser et lier les modèles entre eux. Les modèles concernés regroupent les modèles des besoins et les modèles d'architecture.

Les besoins sont exprimés dans un langage semi-formel au format Requirements Interchange Format (ReqIF). ReqIF (OMG, Requirement Interchange Format (ReqIF), 2016) fournit un schéma XML pour décrire des besoins. Les besoins ainsi formalisés sont ensuite automatiquement convertis en modèle ontologique OWL avant d'être intégré dans une ontologie des besoins, OntReq.

Les modèles d'architecture sont réalisés à l'aide de l'outil MetaGraph 2.0 (Beijing Zhongke Hive Technology Co., 2020) qui permet de construire des modèles à l'aide de l'ontologie GOPPRRE (Graph Object Property Point Relationship Rôle Extension) (Ding, 2021) et du langage KARMA (Ding, 2021). GOPPRRE est une ontologie qui permet de spécifier le méta-méta modèle qui porte le même nom, GOPPRRE. Le langage KARMA est un langage formel qui permet de construire des modèles et des méta-modèles. Ces modèles et méta-modèles sont basés sur le méta-méta-modèle GOPPRRE. Les modèles KARMA construits par l'outil MetaGraph2.0 peuvent être exportés au format OWL.

Pour expliciter les connaissances métier des modèles créés, l'approche proposée dans l'article prévoit d'utiliser une ontologie. Les connaissances métiers sont donc spécifiés par une ontologie basée sur l'ontologie IOF-Core (IOF, 2023). IOF-Core est une ontologie qui spécifie les connaissances relatives aux domaines de l'industrie.

Les trois ontologies précédemment créées sont unifiées au sein d'une même ontologie grâce à une ontologie de haut niveau, Basic Formal Ontology (BFO) (Alan Ruttenberg, 2019).

Pour la vérification des modèles par rapport aux besoins, des jeux de données sont extraits du module architecture de l'ontologie. On effectue ensuite une simulation sur ces jeux de données et un validateur compare les résultats des simulations avec les besoins du module des besoins de l'ontologie. Il existe dans l'ontologie des relations permettant de faire le lien entre un besoin et un composant de l'architecture. Les données du module de l'architecture sont donc liées aux besoins du module des besoins. En fonction des résultats de la comparaison, le validateur va valider ou non les besoins dans l'ontologie. Cette méthode expérimentée par Airbus en 2022 semble pertinente vis-à-vis de la problématique étudiée. Des études ultérieures devront être menées pour savoir si la méthode peut être adaptée au ferroviaire, si ses outils sont acceptables par les acteurs du ferroviaire et s'ils sont interopérables avec les autres outils de la filière ferroviaire.

5 Notre approche

5.1 Introduction de notre approche

Dans la section 4 nous avons posé la problématique à traiter : l'assistance de la vérification par le biais d'approche sémantique. Comme l'illustre la section précédente, des solutions ont déjà été proposées. Cependant elles n'ont pas été adoptées par les acteurs du ferroviaire. Le projet de recherche concerné par cette thèse vise à étudier une approche permettant une interprétation cohérente, complète, unique et équivoque des connaissances impliquées dans la modélisation des systèmes ferroviaires. On vise notamment à améliorer les échanges entre acteurs.

Plus particulièrement, il s'agit d'étudier comment peut être spécifiée une interopérabilité sémantique (Roxin, 2018) entre les modèles conceptuels de données (CDM) ferroviaires, exprimés en UML, et l'architecture fonctionnelle, physique et logique du système.

Pour ce faire, nous considérons les étapes suivantes :

- La formalisation des connaissances métier. C'est-à-dire la conception et l'alignement de (méta-)modèles sémantiques.
- Le choix d'une approche d'interopérabilité (intégré, unifié, fédéré)
- La spécification sous forme de règles logiques DL de contraintes métier, technologiques et opérationnelles liées à l'environnement du système ferroviaire
- La conception de mécanismes de requête optimisés pouvant s'appliquer à(aux) ontologie(s) construites et alignées

Elles sont détaillées dans les sections suivantes.

5.2 Formalisation de connaissances métier

La construction d'ontologie est un procédé manuel long et fastidieux, car les domaines à modéliser sont parfois d'importantes sources de connaissances (et donc des concepts et des relations entre eux). Dans le cas du domaine ferroviaire, tout le système ferroviaire doit être modélisé. Cela nécessite de modéliser le matériel roulant et son infrastructure, l'environnement dans lequel ils sont déployés ainsi que les interactions avec les opérateurs humains (agents de signalisation par exemple). L'infrastructure à elle seule mobilise déjà beaucoup de connaissances pour être modélisé puisqu'elle fait intervenir des connaissances en génie électrique, en télécommunication ou en mécanique.

Les travaux visés par cette thèse s'intéresseront à évaluer les modèles et les ontologies existantes. En effet, des projets ont déjà été développés pour modéliser les connaissances du domaine ferroviaire. Il y a d'abord eu RailML (RailML, 2023) qui représentait les connaissances du domaine ferroviaire en se basant sur RailTopoModel (RTM). RailML propose un ensemble de schéma XML qui modélise les connaissances du ferroviaire. L'objectif de RailML est de proposer un format d'échange de données du ferroviaire en XML pour permettre une interopérabilité sémantique entre les systèmes ferroviaire.

Plus récemment l'Union Internationale des Chemins de fer (UIC) a développé OntoRail (OntoRail,2023). Il s'agit d'une aggrégation d'ontologies du ferroviaire issus de différentes

initiatives (ERA, EULYNX, IFC Rail, RSM, Transmodel, X2Rail-4). OntoRail a été développé dans le cadre du projet Linx4Rail qui vise à adresser les problèmes d'alignement et d'échange de connaissances dans le ferroviaire.

5.3 Choix d'une approche pour l'interopérabilité

Plusieurs définitions d'interopérabilité existent dans la littérature. On reprend ici la définition de l'ISO17261 « Systèmes intelligents de transport – Identification automatique des véhicules et des équipements – Architecture et terminologie du transport intermodal des marchandises » (ISO, 2012) qui définit l'interopérabilité comme la « capacité que possèdent des systèmes à fournir des services et à recevoir des services d'autres systèmes et à utiliser les services ainsi échangés afin de fonctionner ensemble de manière efficace ». En ne considérant que le domaine informatique et en se basant sur (Roxin, 2018), l'interopérabilité peut être implémentée à trois niveaux :

- Physique : Ce niveau concerne le support physique des messages échangés entre les ordinateurs. Les normes hardware (i.e Ethernet) et les normes de la couche basse de l'architecture réseaux ISO/OSI (i.e TCP, IP, http) assurent une interopérabilité à ce niveau.
- Syntaxique : Ce niveau concerne la syntaxe des messages échangés entre ordinateurs. De nombreux standards (i.e XML ou HTML) assurent une interopérabilité à ce niveau.
- Sémantique : Ce niveau concerne la sémantique des messages échangés entre ordinateurs. L'interopérabilité sémantique n'a pas de solution aujourd'hui. Il n'existe pas d'outils standardisés pour avoir une interprétation sémantique universelle des messages. Cependant, les standards du web sémantique permettent de spécifier le sens d'une ressource informatique dans une application donnée pour permettre des traitements automatisés de l'information sur cette ressource. Selon le langage utilisé pour spécifier la sémantique, il existe 3 degrés d'interopérabilité sémantique :
 - Minimale : Cela correspond à la connaissance minimale, celle littéralement transmise par l'énoncé. Ce degré peut être atteint par Ressource Description Framework (RDF).
 - Étendue : Cela correspond à la connaissance minimale nécessaire pour raisonner sur les informations échangées. Ce degré peut être atteint par Ressource Description Framework Schema (RDFS).
 - Total : Cela correspond à la connaissance de ce qui peut être déduit de l'énoncé et de ce qui ne peut pas être déduit de l'énoncé. Il y a une entente entre les agents sur ce qui peut être cru et ce qui ne peut pas être cru.

Indépendamment de la couche considérée, l'interopérabilité peut se faire selon trois approches (ISO:11354-2015) : intégration, unification, fédération. Le choix de l'approche dépend de la proximité des systèmes à interopérer :

- Intégration : Il existe un format commun pour tous les modèles. Ce format doit être univoque et partagé par tous les acteurs (il n'a pas besoin d'être un standard international pour autant). Deux modèles UML sont interopérables par intégration
- Unification : Il existe un format commun à un niveau méta pour tous les modèles. Contrairement à l'approche par intégration, le format commun n'est pas instanciable. Ce format commun fournit des équivalences et des appariements entre les différents formats qu'il instancie pour permettre une correspondance et une interopérabilité entre les systèmes. Deux modèles qui ont pour méta-méta modèle le MOF (Meta Object Facility) sont interopérables par unification
- Fédération : L'interopérabilité se fait à la volé, au cas par cas. Dans cette approche, l'interopérabilité accompagne et ajuste mais n'impose aucune contrainte sur les langages, outils ou méthodes utilisés.

Chacune de ces approches a ses faiblesses (Roxin A. a., 2019) :

- L'approche par intégration est rigide et limite la compatibilité avec des parties externes. Tous les acteurs doivent se mettre d'accord sur un format de données et ne pas en dévier ;

- L'approche par unification ne permet pas une interopérabilité dynamique et impose que les données manipulées soient toutes mappées au même méta-modèle. Il n'est pas possible d'avoir un accord « à la volée » portant sur les appariements ;
- L'approche par fédération est plus difficile à mettre en place et chaque entité impliquée doit fournir des efforts pour travailler efficacement avec les autres entités.

Dans le cadre de ces travaux, on s'intéresse uniquement à l'interopérabilité sémantique. Le choix d'une approche reste à faire. Ce choix doit tenir compte des contraintes du ferroviaire et de l'acceptabilité de ses acteurs. Peuvent-ils tous travailler avec un format commun pour tous les modèles ? Peut-on unifier tous les modèles à un niveau méta ? Ces questions seront étudiées plus en détail dans le cadre des travaux futurs.

5.4 Conception de mécanismes de requête adaptés

Les bases de connaissances sont stockées dans ce qu'on appelle des magasins de triples. Un seul langage permet de les interroger, il s'agit du langage SPARQL (W3C, SPARQL 1.1 Overview, 2013). SPARQL (SPARQL Protocol and RDF Query Language) est un ensemble de spécifications d'un langage de requête et de protocoles pour manipuler les graphes RDF. Il fournit notamment un langage de requête permettant d'interroger des graphes de données RDF. Le langage d'interrogation SPARQL permet d'écrire des requêtes contenant des patterns de triplets où les sujets, objets et prédicats peuvent être des variables. A partir de la version 1.1, SPARQL permet la fédération de requêtes. Selon l'approche d'interopérabilité choisie, il pourrait être pertinent de considérer cette fonctionnalité.

Nous prévoyons d'étudier plus en détails les mécanismes d'interrogation existants et les besoins de la filière ferroviaire quant à l'interrogation de ses données pour concevoir un mécanisme de requêtes adaptées aux besoins de la filière ferroviaire.

6 Conclusion

Dans cet article nous avons présenté le processus de modélisation de système ferroviaire et les principes de l'ingénierie dirigée par les modèles sur lesquels il s'appuie. Nous avons également vu que cette méthodologie comportait des failles dans la communication entre les acteurs multi-métiers, qui n'utilisent pas les mêmes outils ou les mêmes méthodes. Ces différences occasionnent des pertes d'informations et de la génération d'information incohérente. Cela aboutit à la production d'un prototype erroné qui n'est pas conforme aux besoins des utilisateurs. Face à ce problème, apparaît le besoin de créer des outils et des méthodologies pour assister la vérification des modèles produits afin d'éliminer un maximum d'erreurs avant le prototypage. Nous adressons dans cette thèse la problématique de vérification du processus de modélisation par le biais d'approche sémantique.

Pour adresser ces problèmes de communication et d'échange entre les acteurs et, afin d'améliorer les pratiques en ingénierie, plusieurs méthodes et approches ont été proposées. Les méthodes formelles, déjà employées, permettent bien une vérification formelle sur les modèles, mais s'avèrent insuffisantes pour résoudre les problèmes de communication entre acteurs. Une autre approche consiste à annoter les modèles pour créer une connexion entre les modèles et une ontologie du domaine. Cela permet d'avoir une sémantique associée aux modèles et permet le raisonnement formel qui permet la vérification sur les modèles. Cette approche déjà très documentée dans la littérature scientifique n'a pour l'instant pas été adoptée dans le ferroviaire. Cela peut s'expliquer par la complexité des systèmes ferroviaires qui rend ce genre d'approche difficile à utiliser telle quelle. Enfin, récemment, les équipes d'Airbus ont proposé une approche basée sur l'outil MetaGraph 2.0. Cette approche propose d'utiliser une ontologie de haut niveau, BFO, pour travailler unifier les différentes ontologies produites lors du processus de modélisation. De plus, elle utilise l'outil MetaGraph 2.0 qui se base sur le méta-méta-modèle GOPPRE et le langage formel KARMA pour guider la modélisation en intégrant des contraintes de modélisation spécifiées en langage KARMA. Il reste à déterminer si cette méthode peut être adaptée au ferroviaire et si elle est assez expressive pour permettre la communication entre les acteurs et la coopération avec la machine.

Pour les futurs travaux de cette thèse, nous envisageons de poursuivre l'état de l'art sur l'aspect normatif du Building Information Modeling (BIM), en étudiant les concepts de l'Open BIM et les normes promues par buildingSmartInternational (bSI). Nous approfondirons également l'étude des différentes approches pour l'interopérabilité, les implications de chaque approche : intégration, unification, fédération.

Enfin nous étudierons le problème de l'usage des ontologies. À la fois pour leur maintien, leur réutilisabilité mais également pour leur utilisation par des personnes qui ne sont pas expertes en ingénierie des connaissances.

7 Remerciements

Nous remercions l'IRT Railenium qui permet à ces travaux d'exister en finançant cette thèse dans le cadre du projet Interail.

8 Références

- 50126-2:2018. (2018). *AFNOR*. Récupéré sur NF EN 50126-2:2018: <https://www.boutique.afnor.org/fr-fr/norme/nf-en-501262/>
- Alan Ruttenberg, J. Z. (2019). *BFO*. Récupéré sur BFO github: <https://github.com/BFO-ontology/BFO>
- Beijing Zhongke Hive Technology Co., L. (2020). *L'outil de modélisation multi-architecture MetaGraph 2.0 prend en charge l'ingénierie des systèmes complexes*. Récupéré sur L'outil de modélisation multi-architecture MetaGraph 2.0 prend en charge l'ingénierie des systèmes complexes: <https://zhuanlan.zhihu.com/p/422343888>
- Britz, K. a. (2020). Principles of KLM-style defeasible description logics. *ACM Transactions on Computational Logic (TOCL)*, 1-46.
- Bucchiarone, A. a. (2020). Grand challenges in model-driven engineering: an analysis of the state of the research. *Software and Systems Modeling*, 5-13.
- Bunge, M. (1979). *Treatise on Basic Philosophy, in Ontology: A World of Systems*.
- Busard, S. a. (2015). Verification of railway interlocking systems. *arXiv preprint arXiv:1506.03554*.
- CENELEC*. (2023). Récupéré sur CENELEC: <https://www.cenelec.eu/>
- Chouchani, N. a. (2023). ATMO: Autonomous Train Map Ontology. 283-290.
- Chouchani, N. S. (2022). Model-based safety engineering for autonomous train map. *Journal of Systems and Software*.
- D Gašević, D. D. (2006). *Model driven architecture and ontology development*. Springer.
- Davis, A. (2013). *Just enough requirements management: where software development meets marketing*. Addison-Wesley.
- Debbech, S. (2019). Ontologies pour la gestion de sécurité ferroviaire : integration de l'analyse dysfonctionnelle dans la conception.
- Debbech, S. a. (2020). An Ontological Approach to Support Dysfunctional Analysis for Railway System Design. *J. Univers. Comput. Sci*, 549-582.
- Debbech, S. a.-D. (2018). Improving safety by integrating dysfunctional analysis into the design of railway systems. *WIT Transactions on The Built Environment*, 399-411.
- Debbech, S. P.-D. (2019). Towards semantic interpretation of goal-oriented safety decisions based on foundational ontology. *Journal of computers*, 257-267.
- Ding, J. a. (2021). Integration of modeling and verification for system model based on KARMA language. Dans *Proceedings of the 18th ACM SIGPLAN International Workshop on Domain-Specific Modeling* (pp. 41-50).
- Favre, J.-M. (2004). Towards a basic theory to model model driven engineering. *3rd workshop in software model engineering, wisme*, 262-271.
- Guarino, N. a. (1995). Ontologies and knowledge bases. *Towards very large knowledge bases*.
- Hacid, K. a.-A. (2016). *Strengthening MDE and formal design models by references to domain ontologies. A model annotation based approach*. Springer.

- Haxthausen, A. E. (2011). A formal approach for the construction and verification of railway control systems. *Formal Aspects of Computing*, 191-219.
- Hu, X. a. (2022). Ontology-based system to support industrial system design for aircraft assembly. *IFAC-PapersOnLine*, 175-180.
- Hu, X. a. (2022). Ontology-centric industrial requirements validation for aircraft assembly system design. *IFAC-PapersOnLine*, 3016-3021.
- IEEE. (2002). *IEEE explore*. Récupéré sur IEEE Standard Glossary of Software Engineering Terminology: <https://ieeexplore.ieee.org/document/159342/definitions#definitions>
- IEEE. (2023). *ISO/IEC/IEEE International Standard - Systems and Software engineering Methods and tools for model-based systems and software engineering*. Récupéré sur IEEE standards: <https://standards.ieee.org/ieee/24641/10139/>
- IOF, O. (2023). *The IOF Core Ontology*. Récupéré sur IOF-Core github: <https://github.com/iofoundry/ontology/tree/202301/core>
- ISO. (2012). *obb iso.org*. Récupéré sur [iso.org](https://www.iso.org/obp/ui/#iso:std:iso:17261:ed-1:v1:fr): <https://www.iso.org/obp/ui/#iso:std:iso:17261:ed-1:v1:fr>
- ISO. (2015). *iso*. Récupéré sur iso obp: <https://www.iso.org/obp/ui/fr/#iso:std:iso:11354:-2:ed-1:v1:en>
- ITEH. (2017). *Applications ferroviaires - Spécification et démonstration de la fiabilité, de la disponibilité, de la maintenabilité et de la sécurité (FDMS) - Partie 1: Processus FMDS générique*. Récupéré sur [standards iteh](https://standards.iteh.ai/catalog/standards/clc/e5456892-eb2c-437e-8c4b-91c08007f0b4/en-50126-1-2017): <https://standards.iteh.ai/catalog/standards/clc/e5456892-eb2c-437e-8c4b-91c08007f0b4/en-50126-1-2017>
- Kanso, K. a. (2009). Automated verification of signalling principles in railway interlocking systems. *Electronic Notes in Theoretical Computer Science*, 19-31.
- Klaus Pohl, H. H. (2012). *Model-based engineering of embedded systems: The SPES 2020 methodology*. Springer.
- Kleppe, A. G. (2003). *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Professional.
- Kullmann, O. (2008). Present and future of practical SAT solving. *Complexity of Constraints: An Overview of Current Research Themes*, 283-319.
- Leal, G. d. (2019). Interoperability assessment: A systematic literature review. *Computers in Industry*, pp. 111-132.
- Liao, Y. a. (2014). Formal semantic annotations for models interoperability in a PLM environment. *IFAC Proceedings Volumes*, 2382-2393.
- Nielsen, C. B. (2015). Systems of systems engineering: basic concepts, model-based techniques, and research directions. *ACM Computing Surveys (CSUR)*, 1-41.
- OMG. (2003). *OMG Unified Modeling Language Specification*.
- OMG. (2016). *Requirement Interchange Format (ReqIF)*. Récupéré sur [omg.org](https://www.omg.org/spec/ReqIF/1.2/PDF): <https://www.omg.org/spec/ReqIF/1.2/PDF>
- OMG. (2019). *OMG Meta Object Facility (MOF) Core Specification*. Récupéré sur [omg.org](https://www.omg.org).
- OntoRail. (s.d.). *ontorail*. Récupéré sur [ontorail](http://ontorail.brainz.ai:5000/ontorail): <http://ontorail.brainz.ai:5000/ontorail>
- Rahmig, C. a. (2022). Ontology-based Conceptual Model Development for the Railway Domain: A Maintenance Case Study.
- RailML. (2023). *railML.org*. Récupéré sur [railML.org](https://www.railml.org/en/): <https://www.railml.org/en/>
- Roxin, A. (2018). Reasoning and knowledge - tradeoff between expressivity and efficiency.
- Roxin, A. a. (2019). Semantic interoperability between BIM and GIS--review of existing standards and depiction of a novel approach. *36th CIB W78--Information Technology for Construction*.
- Seidewitz, E. (2003). What models mean. *IEEE Software*, 26-32.
- Studer, R. a. (1998). Knowledge engineering: Principles and methods. *Data & knowledge engineering*, 161-197.
- Truyen, F. (2006). The fast guide to model driven architecture the basics of model. *Cephas Consulting Corp*.

- Truyen, F. (2006). The fast guide to model driven architecture the basics of model driven architecture. *Cephas Consulting Corp.*
- Tu, Z. a. (2016). A federated approach to develop enterprise interoperability. *Journal of Intelligent Manufacturing*, pp. 11-31.
- Van Harmelen, F. a. (2008). *Handbook of knowledge representation*. Elsevier.
- Voirin, J.-L. a. (2016). Simplifying (and enriching) SysML to perform functional analysis and model instances. *INCOSE International Symposium*, 253-268.
- W3C. (2013). *SPARQL 1.1 Overview*. Récupéré sur w3c.org: <https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/>
- W3C. (2013). *SPARQL 1.1 Query Language*. Récupéré sur w3c.org: <https://www.w3.org/TR/sparql11-query/>
- Walter Schön, G. L. (2013). *Signalisation et automatismes ferroviaires-Tome 1, Vie du Rail*.
- Yahia, E. a. (2012). Semantics enactment for interoperability assessment in Enterprise Information Systems. *Annual Reviews in Control*, 106-117.