



HAL
open science

Etude comparative des méthodes de création des modèles de transformation

Quentin Esma, Lionel Roucoules, Esma Yahia, Lionel Roucoules

► **To cite this version:**

Quentin Esma, Lionel Roucoules, Esma Yahia, Lionel Roucoules. Etude comparative des méthodes de création des modèles de transformation. S-MART 2023 : 18ème Colloque national S.mart, Arts et Métiers Paristech ENSAM Aix-en-Provence, Université de Toulon [UTLN], Apr 2023, Carry-le-Rouet, France. hal-04278705

HAL Id: hal-04278705

<https://hal.science/hal-04278705v1>

Submitted on 10 Nov 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.

Etude comparative des méthodes de création des modèles de transformation

Q. Brilhault ^{a,*}, E. Yahia ^a, L. Roucoules ^a

^a Arts et Métiers Institut de Technologie, HESAM Université, LISPEN, 13617 Aix-en-Provence, France

* e-mail : quentin.brilhault@ensam.eu

1. INTRODUCTION

La prédisposition des systèmes à communiquer, échanger de l'information puis utiliser les informations échangées fait appel à la notion d'interopérabilité des systèmes [1]. La nécessité d'instaurer un climat d'interopérabilité dans les entreprises n'est plus à démontrer. La coopération entre les systèmes (machine, robot, objets connectés, système d'information...) à travers le partage et l'échange d'informations permet de résoudre des activités complexes qu'un système seul ne pourrait résoudre individuellement.

Néanmoins, l'état actuel de l'environnement numérique industriel prouve qu'il est délicat de répondre aux exigences d'hyperconnectivité des technologies de l'Industrie du Futur. Les systèmes sont pour la plupart isolés, incompatibles et de plus en plus complexes. Leur conception emploie différents langages de programmation, divers protocoles de communication et un certain nombre de formats de données d'échanges qui inhibent l'interopérabilité des systèmes au niveau technique. Au niveau sémantique, l'opération de mappage qui vise à établir des relations d'équivalence entre les différentes représentations conceptuelles d'une même donnée est une activité délicate. L'hétérogénéité des systèmes impose impérativement le développement d'adaptateurs *ad-hoc* propres à chaque type de communications pour gommer les différences des systèmes. Or, dans un environnement numérique instable et dynamique bercé par l'émergence de nouvelles technologies, par la prolifération des nouvelles sources d'information et par la nécessité d'accroître la flexibilité et l'évolutivité de la chaîne de valeur, ces adaptateurs statiques semblent peu performants.

L'un des principaux défis pour l'interopérabilité du futur consiste à développer des connecteurs durables capables d'absorber de futurs changements ou recommandations pour s'adapter aux spécifications des

systèmes à interconnecter [2], et ce, en minimisant l'intervention humaine.

Les techniques de transformations de modèles pourraient fournir une solution concrète pour répondre aux exigences d'une interopérabilité durable. Les techniques de transformation de modèles constituent la pierre angulaire de l'Architecture Dirigée par les Modèles [3] (*Model-Driven Architecture MDA*). L'architecture MDA définit comment un modèle, conforme à son métamodèle, peut être transformé en un modèle conforme à un tout autre métamodèle [4] (voir Figure 1).

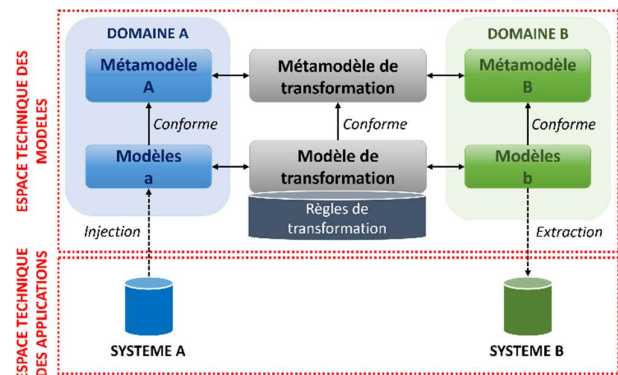


Figure 1 - Architecture Dirigée par les Modèles

Les techniques de transformation de modèle facilitent l'interopérabilité entre les modèles [5], et, *in fine*, elles permettent l'interopérabilité des systèmes. De nombreuses approches développées ces dernières années ont pour objectif de réduire les efforts à mettre en œuvre pour construire de manière automatique ou semi-automatique ces modèles de transformation. Cependant, les approches d'interopérabilité basées sur les modèles de transformation manquent d'un réel cadre expérimental propice à l'évaluation et à la comparaison des nouvelles approches vis-à-vis de la littérature existante. Les approches ne sont pas toujours validées par les mêmes indicateurs de performances et ne sont pas

souvent appliquées aux mêmes jeux de données. Compte tenu de l'hétérogénéité des métamodèles, il n'est pas toujours aisé de comparer les approches entre-elles puisqu'un modèle de transformation peut faire intervenir une multitude de patterns de transformation.

Suivant la méthode présentée en section 2, un protocole expérimental, permettant la comparaison des différentes approches d'interopérabilité des systèmes basées sur les modèles de transformation, sera proposé en section 3. La section 4 présentera quant à elle une étude comparative des différentes approches existantes répertoriées dans la littérature. La section 5 sera consacrée à la discussion des résultats obtenus par l'étude précédente.

2. METHODE

Dans le cadre d'une étude comparative, ou « *benchmark* », la construction d'un protocole expérimental est une activité rigoureuse qui nécessite de définir un certain nombre de paramètres. La Figure 2 présente l'agencement de ces paramètres les uns par rapport aux autres sous forme de métamodèle et peut être retrouvé via le *GitHub* <https://github.com/GIS-S-mart>.

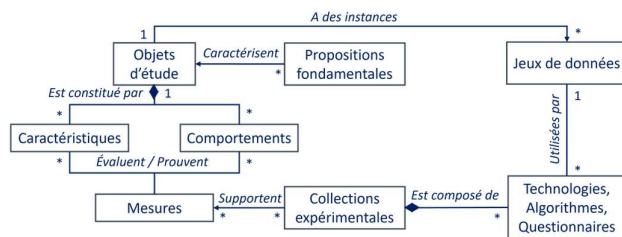


Figure 2 – Métamodélisation de la structure d'une étude comparative type « *benchmark* »

Proposition fondamentale. Cette étude repose sur l'hypothèse que les récentes avancées dans les domaines de l'apprentissage machine et de l'apprentissage profond simplifient et optimisent la création de modèles de transformation.

Objet d'étude. L'objet d'étude portera sur les méthodes permettant l'écriture des modèles de transformation.

Caractéristiques. Ces modèles de transformation sont définis par un ensemble de règles qui permettent la transformation d'un pattern source en un pattern cible. Ainsi les caractéristiques de l'objet d'étude correspondent aux différents patterns de transformation qui peuvent intervenir lors d'une transformation de modèle.

Mesures. Des indicateurs de performance ont été définis pour évaluer la capacité des différentes approches à inférer le modèle de transformation attendu.

Collections expérimentales. Une collection expérimentale recense l'ensemble des approches, ou outils (algorithmes, technologies...) consacrés à l'objet d'étude.

Jeux de données. Chacune des approches de la collection expérimentale sera appliquée à des jeux de données spécifiques permettant de valider, selon les

mesures de performance, la capacité des approches à résoudre certains patterns de transformation.

3. PROTOCOLE EXPERIMENTAL

3.1. Objet d'étude : les méthodes d'écriture des modèles de transformation

L'ingénierie dirigée par les modèles (« *Model-Driven Engineering* » MDE) favorise l'utilisation de « modèles » comme concept unificateur pour représenter un objet, un système ou encore une partie d'un système [6]. Selon ses principes, un modèle (M1) est une représentation abstraite de la réalité, tandis que le métamodèle (M2), ou langage de modélisation, est quant à lui le langage qui décrit comment un objet du monde réel est modélisé d'un point de vue structurel et terminologique. De cette façon, un système complexe peut être décomposé en plusieurs métamodèles, ou points de vue, qui procurent une abstraction pour chacune des préoccupations désirées.

Le modèle de transformation est l'élément central d'une architecture MDA. Il est défini par un ensemble de relations d'équivalence qui relient les concepts d'un métamodèle source aux concepts d'un métamodèle cible. Ces relations sont appelées règles de transformation et remplissent la fonction d'interopérabilité structurelle et sémantique entre les différents modèles de données par l'intermédiaire de leur métamodèle [7].

Suivant la Figure 1, les systèmes A et B sont respectivement méta-modélisés par les métamodèles A et B qui procurent la sémantique et la structure des données des deux systèmes. Le système A initie une communication avec le système B par l'intermédiaire de l'espace technique des modèles. Les données du système A sont alors injectées au sein d'un modèle « a » qui sera ensuite transformé suivant les règles de transformation définies par le modèle de transformation. La transformation du modèle « a » aboutira à la création d'un modèle « b » conforme à la sémantique et à la structure du métamodèle B. Les données du modèle « b » seront ensuite extraites puis intégrées au sein du système B.

3.2. Caractéristiques de l'objet d'étude : les patterns de transformation

Garantir l'interopérabilité entre les modèles (M1) est une tâche ardue. La mise en relation des concepts provenant de différents métamodèles (M2) est nécessaire. La première étape lorsqu'il s'agit d'établir une transformation entre deux métamodèles est d'identifier les relations de correspondance (i.e. le mappage) entre les éléments des métamodèles (i.e. classes, associations et attributs).

Or, la prolifération des langages de modélisation complexifie grandement leurs capacités à s'interconnecter convenablement et a pour conséquence de pérenniser et d'étendre un écosystème composé de modèles hétérogènes. De nombreux modèles existent et chacun d'eux possède une structure de modélisation et une terminologie particulière entraînant une hétérogénéité sémantique des métamodèles.

L'hétérogénéité sémantique est avant tout provoquée par des perspectives multiples, i.e. des points de vue différents de la donnée [8]. Par conséquent, selon le point de vue, deux concepts sémantiquement similaires peuvent être méta-modélisés (M2) par des structures et des terminologies qui diffèrent selon les domaines. D'autre part, au niveau modèle (M1), la valeur des attributs (ou data) peut, elle aussi, être sujette à une hétérogénéité syntaxique. La valeur d'un attribut peut être écrite de différentes façons (i.e. abréviation, arrondi, conversion...) et complexifie davantage l'identification des relations de correspondance.

Pour mesurer toute la complexité lorsqu'il s'agit d'éditer une transformation de modèle, une compréhension plus approfondie des conflits à résoudre provoqués par l'hétérogénéité des métamodèles est nécessaire. L'étude de ces conflits permet de caractériser les patterns de transformation qui interviennent lors d'une transformation de modèle.

Dans leur article, [9] proposent une classification des types de conflits en deux grandes catégories (1) les conflits de type sémantiques où la terminologie des concepts et la valeur des attributs sont sujettes à de mauvaises interprétations ; (2) les conflits de type structuraux. Dans cette étude, les conflits sémantiques seront classifiés sous forme de conflits (1) terminologiques et syntaxiques ; (2) structuraux.

Les conflits structuraux. Selon les métamodèles (M2), une même information peut être représentée par différents concepts. Bien que deux structures de modélisation soient différentes, elles peuvent tout de même véhiculer la même information. Les conflits structuraux peuvent être eux même catégorisés selon deux points de vue :

- Un *conflit structurel interne* se traduit par deux concepts sémantiquement similaires qui ne disposent pas, en totalité ou en partie, des mêmes attributs.
- Un *conflit structurel externe* implique que deux représentations sémantiquement similaires peuvent être entourées par des structures adjacentes (i.e. relation et classes voisines) non-isomorphiques.

Les différents patterns de transformation sont répertoriés dans le Tableau 1.

Tableau 1 - Patterns de transformation

Pattern	Description
[1-1]	Un concept du métamodèle source est transformé en un concept du métamodèle cible.
[n-1] [1-n]	Plusieurs concepts du métamodèle source sont transformés en un unique concept du métamodèle cible. Inversement pour le pattern [1-n].
[n-m]	Plusieurs concepts du métamodèle source sont transformés en plusieurs concepts du métamodèle cible.

Par ailleurs, certains patterns de transformation peuvent être soumis à des conditions pouvant orienter la transformation d'un concept ou d'un groupe de concepts. Nous pouvons alors émettre l'hypothèse que la

transformation est guidée par des caractéristiques dites propres aux concepts.

Tableau 2 - Conditions de transformation

Symbole	Types de condition	Description
σ_r	Condition sur une relation	La transformation d'une classe peut être conditionnée par une ou plusieurs de ses relations.
σ_a	Condition sur la valeur d'un attribut de la classe	La transformation d'une classe peut être conditionnée par la valeur d'un ou plusieurs de ses attributs.
σ_c	Condition sur les classes adjacentes d'ordre 1	La transformation d'une classe peut être conditionnée par une ou plusieurs de ses classes voisines (type de classe ou bien valeur spécifique d'un attribut)

Les conflits terminologiques. Au niveau métamodèle (M2), les conflits sémantiques lors du nommage des éléments sont provoqués par l'utilisation de synonymes, d'antonymes ou encore par des granularités différentes lors de la modélisation des données qui font intervenir des hyponymes et des hyperonymes. Des conflits syntaxiques peuvent également intervenir lors de l'utilisation d'abréviations et de conventions d'écriture. Au niveau modèle (M1) la valeur des attributs peut engendrer des conflits sémantiques dans le cas d'opérations mathématiques (conversions des devises, changement d'unités de valeur), ou encore lors d'approximations d'une valeur numérique (arrondies). Des conflits syntaxiques peuvent être induits dans le cas d'une concaténation de valeurs (deux attributs forment un unique attribut) et de formats de données différents (conversion d'une date au format anglais).

Classification des conflits sémantiques. La classification proposée n'est pas exhaustive et mérite d'être complétée dans le cadre de futures expérimentations. Cette classification à vocation à servir de socle commun pour caractériser les transformations de modèle. La Figure 3 présente les différents types de conflits répertoriés, à savoir, les caractéristiques qui définissent un modèle de transformation.

Une grille d'évaluation a été construite pour caractériser les modèles de transformation vis-à-vis des conflits structuraux et terminologiques. Son objectif est de simplifier la comparaison des différentes approches en introduisant un cadre de référence qui répertorie les différents patterns de transformation. Le Tableau 3 et le Tableau 4 présentent un échantillon des différents patterns de transformation possibles.

Tableau 3 - Exemple de conflits structuraux

Types	N°	Pattern	Pattern Source	Pattern Cible
Concaténation de classes	(c2)	[n-1]	$\begin{array}{ c } \hline c_1 \\ \hline a_1 = v_1 \\ \hline \end{array} \xrightarrow{r} \begin{array}{ c } \hline c_2 \\ \hline a_2 = v_2 \\ \hline \end{array}$	$\xrightarrow{f} \begin{array}{ c } \hline c'_1 \\ \hline a'_1 = v_1 \\ \hline a'_2 = v_2 \\ \hline \end{array}$
Concaténation d'attributs	(a2)	[n-1]	$\begin{array}{ c } \hline c_1 \\ \hline a_1 = v_1 \\ \hline \end{array} \xrightarrow{r} \begin{array}{ c } \hline c_2 \\ \hline a_2 = v_2 \\ \hline \end{array}$	$\xrightarrow{f} \begin{array}{ c } \hline c'_1 \\ \hline a'_1 = v_1 + v_2 \\ \hline \end{array}$

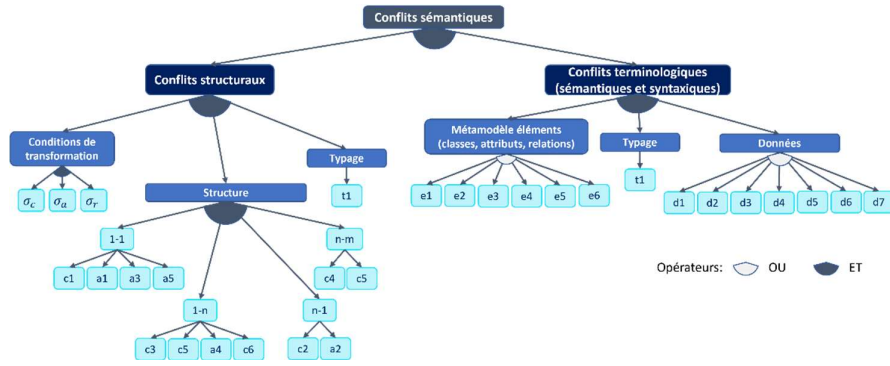


Figure 3 - Classification des conflits sémantiques

Tableau 4 - Exemple de conflits terminologiques et syntaxiques

Eléments	Type	N°
Eléments (M2)	Granularité	(e4)
	Champs lexical/Linguistique	(e6)
Données (M1)	Texte descriptif	(d3)
	Opération mathématique	(d4)

3.3. Jeux de données de validation

Dans le cadre des expérimentations pour tester la validité des approches à résoudre certains types de patterns de transformation, des jeux de données ont été identifiés. Les métamodèles des transformations *Families2Persons* et *Class2Relational* du Zoo ATL mis à disposition par la fondation Eclipse ont été utilisés. La transformation *City2TownHall* constituera le troisième jeu de données de validation. Cette dernière a la particularité d'être une transformation bidirectionnelle, i.e. la transformation du métamodèle *Town Hall* vers le métamodèle *City* est possible.

- La transformation *Families2Persons* impose une compréhension des associations qu'entretiennent les classes entre elles. Le seul moyen d'assurer la transformation est de capturer la signification des associations de chacune des classes. De ce fait, il est nécessaire de prendre en compte, non seulement la classe, mais également les types d'associations qu'elles détiennent pour capturer les conditions en relation σ_r .
- La transformation *Class2Relational* nécessite une compréhension de la structure interne des classes. La valeur d'un attribut conditionne la transformation d'une classe (σ_a). Il est alors important de différencier les classes du même type en fonction des attributs qu'elles détiennent.
- La transformation *City2TownHall* et *TownHall2City*, impose une compréhension de la structure externe des classes. La valeur d'un attribut d'une des classes adjacentes à la classe étudiée conditionne sa transformation (σ_c). Il est alors important de différencier les classes du même type en fonction des relations

qu'elles entretiennent avec les classes adjacentes.

Le Tableau 5 et le Tableau 6 synthétisent la caractérisation des différentes transformations utilisées à titre de jeux de données de validation.

Tableau 5 - Caractérisation des conflits structuraux des jeux de données

Transformation	Conflits structuraux						
	Patterns de transformation				Conditions de transformation		
	1-1	1-n	n-1	n-m	σ_c	σ_a	σ_r
<i>Families2Persons</i>	-	-	(c2)	-	-	-	x
<i>Class2Relational</i>	(c1)	(c5)	-	(c4)	-	x	x
<i>City2TownHall</i>	(c1)	-	(c2)	(c4)	x	-	x
<i>TownHall2City</i>	(c1)	(c5)	(c2)	(c6)	x	-	x

Tableau 6 - Caractérisation des conflits terminologiques des jeux de données

Transformation	Conflits terminologiques			
	Classes	Relations	Attributs	Données
<i>Families2Persons</i>	(e4)	(e4)	(e3)	-
<i>Class2Relational</i>	(e6)	(e6)	(e6)	-
<i>City2TownHall</i>	(e4)	(e4)(e3)	(e3)	-
<i>TownHall2City</i>	(e4)	(e4)(e3)	(e3)	-

3.4. Mesures de performance

Des critères de performance ont été identifiés pour mesurer la capacité des approches à résoudre certains types de patterns de transformation. Les mesures de performances suivantes ont été définies :

Pour la phase d'apprentissage :

- Comme l'objectif d'un modèle de transformation est de minimiser les pertes d'informations lors de la conversion d'un modèle, des métriques telles que le *recall* (R), la précision (P) et la f-mesure ont été utilisées pour mesurer la validité des modèles obtenus après transformation. Ces premiers indicateurs permettent d'évaluer si les règles de

transformation ont correctement été apprises lors de la phase d'apprentissage.

$$F_{\text{mesure}} = 2 \frac{P * R}{P + R} \quad (1)$$

- (2) Les efforts humains :
- En termes de pré-traitement par rapport aux éléments à fournir pour parfaire l'apprentissage des règles de transformation, à savoir les données d'exploitation (le nombre de modèles à fournir ainsi que les éléments complémentaires tel que des pré-alignements).
 - En termes de post-traitement et de validation des règles inférées.
- (3) La durée d'entraînement nécessaire pour apprendre les règles de transformation.

Par rapport à la phase de prédiction, les métriques telle que le *recall*, la précision et la f-mesure seront réutilisées pour vérifier la capacité des règles de transformation précédemment inférées à transformer tout nouveau modèle conforme au métamodèle appris.

4. BENCHMARK

4.1. Collection expérimentale : état de l'art

Lorsqu'il s'agit d'écrire une transformation de modèle, un mappage entre les éléments d'un métamodèle source et d'un métamodèle cible est nécessaire. Deux catégories d'approches tentent d'automatiser la génération de ce mappage. La première est basée sur les éléments du schéma (métamodèle M2), tandis que la seconde approche est basée sur l'étude des instances (modèles M1), à savoir l'analyse des valeurs des attributs.

Dans cet article nous porterons une attention toute particulière aux approches d'apprentissage des règles de transformation par l'exemple initiée par Varró [10], i.e. qui s'appuient sur des instances de modèles source et cible données en entraînement. Le Tableau 7 répertorie l'ensemble des approches d'apprentissage par l'exemple des modèles de transformation.

Tableau 7 - Collection expérimentale d'algorithme d'apprentissage par l'exemple des modèles de transformation

Approches	Algorithmes	Entrées	Sorties
[11]	Programmation Logique Inductive	Modèles source/cible Pré-alignement	Règles
[12][13]	RCA	Modèles source/cible Pré-alignement	Règles
[14]	Programmation Génétique	Modèles source/cible	Règles
[15]	Programmation Génétique	Modèles source/cible Pré-alignement	Règles
[16]	LSTM Encoder/Decoder	Modèles source/cible	Modèle cible
[17]	Apprentissage Q par renforcement	Modèles source/cible	Règles / Modèle cible

4.2. Résultats

L'étude comparative menée suit le protocole expérimental établi en section 3 qui a lui-même été appliqué à la collection expérimentale présentée en 4.1. Le jeu de données sélectionné pour réaliser l'étude comparative de cet article est la transformation *Class2Relational*.

Le Tableau 8 synthétise les résultats obtenus lors de l'étude, à savoir la capacité de chacune des approches à pouvoir inférer le modèle de transformation permettant de transformer une instance de modèle de classe en instance de modèle relationnel. Les métamodèles et modèles utilisés ont été construits à partir de d'Eclipse Modeling Framework (EMF) et décrits en suivant le formalisme du méta-métamodèle Ecore.

Tableau 8 - Etude comparative des approches d'apprentissage des règles de transformation par l'exemple

Approches	Performances de l'apprentissage			Perf. Prédiction	Niveau	
	Temps (s)	P	R	F		
[11]	?	?	?	?	0	
[12][13]	?	0.9	0.8	0.85	0.93	1
[14]	?	?	0.75	?	?	2
[15]	?	0.94	0.94	0.94	0.94	1
[16]	±1000	1.0	1.0	1.0	1.0	3
[17]	2.5	1.0	1.0	1.0	1.0	3

L'étude précédente suit 4 niveaux d'analyse :

- 0 : les résultats sont difficilement exploitables ou ne sont pas clairement définis.
- 1 : les résultats sont exploitables mais des calculs ont été réalisés.
- 2 : les résultats ont été repris tel qu'ils sont présentés dans l'article.
- 3 : les algorithmes ont été exécutés pour obtenir les résultats.

5. DISCUSSION

5.1. Un pré-alignement manuel entre les modèles source et cible

La plupart des travaux exploitent un pré-alignement, i.e. des relations de correspondances manuellement préétablies entre les éléments des modèles source et cible. Ce pré-alignement n'est pas toujours évident à fournir, il nécessite souvent un heuristique métier. D'autre part, comme souligné par Burgueño et al. [16] la majeure partie des propositions vise à simplifier la définition des modèles de transformation sans pour autant totalement l'automatiser. Force est de constater que l'impact de l'humain est omniprésent dans le processus de description des modèles de transformation.

5.2. Une application des réseaux de neurones

Malgré les résultats très prometteurs présentés par Burgueño et al [16], une approche basée sur l'apprentissage des modèles de transformation par un réseau de neurones semble inapplicable dans le cadre de

l'Industrie 4.0. D'un point de vue technique, un réseau de neurones artificiels requiert un grand nombre de données pour parfaire son entraînement (750 modèles source et cible sont nécessaires pour obtenir une prédiction précise à 100% dans le cas de la transformation *Class2Relational*). D'autant plus que les mappages complexes soumis à des conditions et de types n-m nécessitent davantage de données d'entraînement. La diversité du jeu d'entraînement est un autre facteur limitant qui porte préjudice à l'approche énoncée par Burgueño et al. Le réseau de neurones artificiels est capable de prédire uniquement les scénarios qu'il a auparavant appris. Enfin la durée de l'entraînement dépend grandement du nombre de modèles (pour 4000 modèles la durée de l'entraînement est de 483 secondes), mais aussi du nombre d'éléments par modèles (pour 30 éléments dans un modèle la durée de l'entraînement est de 4060 secondes).

5.3. Une approche par renforcement

Comme l'affirme Mnih et al. la performance d'un apprentissage par renforcement repose essentiellement sur la capacité à décrire les caractéristiques qui représentent l'environnement [18]. Dans l'étude [17] chaque classe est représentée par sa forme aplatie pour capturer sa structure interne et adjacente. Cette représentation semble beaucoup trop simpliste lorsqu'il s'agit de tenir compte de dépendances plus lointaines entre les classes (supérieure à l'ordre 2). Par ailleurs, la fonction d'extraction de la représentation des classes est spécifiée par le développeur. De ce fait, de nombreuses caractéristiques essentielles ne sont pas capturées lorsqu'il s'agit d'établir une représentation fidèle des classes (par exemple l'orientation des relations entre les classes...).

6. CONCLUSION

Pour répondre aux défis de l'interopérabilité du futur, une approche d'Interopérabilité Dirigée par les Modèles basée sur les modèles de transformation semble être une solution viable. De nombreuses propositions visant l'automatisation des modèles de transformation ont vu le jour ces dernières années. Cependant ces approches souffrent d'un réel cadre permettant leur comparaison.

Dans cette étude, un protocole expérimental est proposé à la communauté scientifique pour évaluer les nouvelles propositions vis-à-vis des solutions existantes. L'objet d'étude porte sur les modèles de transformation et les méthodes permettant de simplifier l'édition des règles de transformation. Ce protocole contient, des jeux de données, des critères de validation, ainsi qu'une grille d'évaluation pour caractériser les différents patterns de transformation qui interviennent pendant une transformation de modèle.

Une étude comparative des approches existantes a été réalisée suivant le protocole expérimental présenté dans cet article. La discussion des résultats aboutit sur l'importance d'évaluer les efforts humains nécessaire à

fournir lorsqu'il s'agit d'éditer un modèle de transformation en utilisant chacune des approches présentées dans la collection expérimentale. De ce fait, le ratio entre les efforts humains (éléments à fournir à l'algorithme) et la performance finale de l'algorithme est à évaluer.

REMERCIEMENTS

Les auteurs souhaitent remercier le Centre National d'Etudes Spatiales (CNES) ainsi que Thales Alenia Space pour le soutien humain, matériel et financier apportés à leurs travaux.

RÉFÉRENCES

- [1] D. Chen, « Framework for Enterprise Interoperability », *Enterp. Interoperability INTEROP-PGSO Vis.*, vol. 1, p. 8, 2017.
- [2] C. Agostinho et al., « Towards a sustainable interoperability in networked enterprise information systems: Trends of knowledge and model-driven technology », *Comput. Ind.*, vol. 79, p. 64-76, juin 2016, doi: 10.1016/j.compind.2015.07.001.
- [3] R. O. M. G. Soley, « Model Driven Architecture », *Object Manag. Group*, p. 12, 2000.
- [4] J. Warmer et A. Kleppe, *The object constraint language: getting your models ready for MDA*, Addison-Wesley Professional, 2003.
- [5] G. Zacharewicz, N. Daclin, G. Doumeings, et H. Haidar, « Model Driven Interoperability for System Engineering », p. 28, 2020.
- [6] J. Bézin, « On the unification power of models », *Softw. Syst. Model.*, vol. 4, n° 2, p. 171-188, 2005.
- [7] M. L. Zeng, « Interoperability », *KO Knowl. Organ.*, vol. 46, n° 2, p. 122-146, 2019, doi: doi.org/10.5771/0943-7444-2019-2-122.
- [8] F. Zenker et P. Gärdenfors, Éd., *Applications of Conceptual Spaces*. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-15021-5.
- [9] M. Wimmer, G. Kappel, A. Kusel, W. Retschitzegger, J. Schoenboeck, et W. Schwinger, « From the Heterogeneity Jungle to Systematic Benchmarking », in *Models in Software Engineering*, vol. 6627, J. Dingel et A. Solberg, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, p. 150-164. doi: 10.1007/978-3-642-21210-9_15.
- [10] D. Varró, « Model Transformation By Example », *Model Driven Eng. Lang. Syst.*, p. 15, 2006.
- [11] Z. Balogh et D. Varró, « Model transformation by example using inductive logic programming », *Softw. Syst. Model.*, p. 18, 2009.
- [12] X. Dolques, M. Huchard, C. Nebut, et P. Reitz, « Learning transformation rules from transformation examples: An approach based on Relational Concept Analysis », *14th IEEE Int. Enterp. Distrib. Object Comput. Conf. Workshop*, p. 7, 2010.
- [13] H. Saada, X. Dolques, M. Huchard, C. Nebut, et H. Sahrroui, « Generation of Operational Transformation Rules from Examples of Model Transformations », in *Model Driven Engineering Languages and Systems*, vol. 7590, R. B. France, J. Kazmeier, R. Breu, et C. Atkinson, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, p. 546-561. doi: 10.1007/978-3-642-33666-9_35.
- [14] M. Faunes, H. Sahrroui, et M. Boukadoum, « Generating model transformation rules from examples using an evolutionary algorithm », in *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering - ASE 2012*, Essen, Germany, 2012, p. 250. doi: 10.1145/2351676.2351714.
- [15] I. Baki et H. Sahrroui, « Multi-Step Learning and Adaptive Search for Learning Complex Model Transformations from Examples », *ACM Trans. Softw. Eng. Methodol.*, vol. 25, n° 3, p. 1-37, août 2016, doi: 10.1145/2904904.
- [16] L. Burgueño, J. Cabot, S. Li, et S. Gérard, « A generic LSTM neural network architecture to infer heterogeneous model transformations », *Softw. Syst. Model.*, mai 2021, doi: 10.1007/s10270-021-00893-y.
- [17] Q. Brilhault, E. Yahia, et L. Roucoules, « Digital continuity based on reinforcement learning model transformations », présenté à International Joint Conference on Mechanics, Design, Engineering and Advanced Manufacturing, Ischia, Italie, juin 2022, p. 12.
- [18] V. Mnih et al., « Playing Atari with Deep Reinforcement Learning ». arXiv, 19 décembre 2013. Consulté le: 13 septembre 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1312.5602>