



HAL
open science

Modélisation sémantique des bases de données d'inventaires en cycle de vie

Benjamin Bertin

► **To cite this version:**

Benjamin Bertin. Modélisation sémantique des bases de données d'inventaires en cycle de vie. Autre [cs.OH]. INSA de Lyon, 2013. Français. NNT : 2013ISAL0049 . tel-00876636

HAL Id: tel-00876636

<https://theses.hal.science/tel-00876636v1>

Submitted on 25 Oct 2013

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.

THÈSE

présentée devant
L'Institut National des Sciences Appliquées de Lyon

pour obtenir le grade de
Docteur

spécialité
Informatique

par
Benjamin Bertin

MODÉLISATION SÉMANTIQUE DES BASES DE DONNÉES D'INVENTAIRES EN CYCLE DE VIE

Soutenue le 25 juin 2013 devant le jury :

Dominique RIEU	Professeur des Universités	Université de Grenoble	Rapporteur
Ioan TODINCA	Professeur des Universités	Université d'Orléans	Rapporteur
Christine LARGERON	Professeur des Universités	Université Jean-Monnet	Examinatrice
Bruno DEFUDE	Professeur	Télécom SudParis	Examineur
Jean-Marie PINON	Professeur des Universités	INSA de Lyon	Co-Directeur
Vasile Marian SCUTURICI	Maître de Conférences	INSA de Lyon	Co-directeur
Emmanuel RISLER	Directeur scientifique	My C-Sense	Encadrant industriel

My C-SENSE
UNIVERSITÉ DE LYON
LABORATOIRE D'INFORMATIQUE EN IMAGE ET SYSTÈMES D'INFORMATION
ÉCOLE DOCTORALE INFORMATIQUE ET MATHÉMATIQUES

Benjamin Bertin : *Modélisation sémantique des bases de données d'inventaires en cycle de vie*, Thèse de Doctorat © 2013

DIRECTEURS DE THÈSE :

Jean-Marie Pinon

Vasile Marian Scuturici

PÉRIODE :

Décembre 2009 – Juin 2013

RÉSUMÉ

L'analyse des impacts environnementaux de la production de biens et de services est aujourd'hui devenue un enjeu majeur. L'analyse en cycle de vie est la méthode consacrée pour modéliser les impacts environnementaux des activités humaines. L'inventaire en cycle de vie, qui est l'une des étapes de cette méthode, consiste à décomposer les activités économiques en processus interdépendants. Chaque processus a des impacts environnementaux et la composition de ces processus nous donne l'impact cumulé des activités étudiées. Plusieurs entreprises et agences gouvernementales fournissent des bases de données d'inventaires en cycle de vie pour que les experts puissent réutiliser des processus déjà étudiés lors de l'analyse d'un nouveau système.

L'audit et la compréhension de ces inventaires nécessite de s'intéresser à un très grand nombre de processus et à leurs relations d'interdépendance. Ces bases de données peuvent comporter plusieurs milliers de processus et des dizaines de milliers de relations de dépendance. Pour les experts qui utilisent les bases de données d'inventaire en cycle de vie, deux problèmes importants sont clairement identifiés :

- organiser les processus pour avoir une meilleure compréhensibilité du modèle ;
- calculer les impacts d'une modélisation (composition de processus) et, le cas échéant, détecter les raisons de la non convergence du calcul.

Dans cette thèse, nous :

- mettons en évidence de l'existence de similarités sémantiques entre les processus et leurs relations d'interdépendance et proposons une nouvelle approche pour modéliser les relations d'interdépendance entre les processus d'une base de données d'inventaire. Elle se base sur un étiquetage sémantique des processus à l'aide d'une ontologie et une modélisation multi-niveaux des relations d'interdépendance entre les processus. Nous étudions aussi deux approches déclaratives d'interaction avec ce modèle multi-niveau.
- étudions les différentes méthodes de calcul des impacts basées sur des notions classiques d'algèbre linéaire et de théorie des graphes. Nous étudions aussi les conditions de non convergence de ces méthodes en présence de cycle dans le modèle des relations de dépendances.

Un prototype implémentant cette approche a montré des résultats probants sur les cas étudiés. Nous avons réalisé une étude de cas de ce prototype sur les processus de production d'électricité aux États-Unis extraits de la base de données d'inventaire en cycle de vie de l'agence environnementale américaine. Ce prototype est à la base d'une application opérationnelle utilisée par l'entreprise.

ABSTRACT

Environmental impact assessment of goods and services is nowadays a major challenge for both economic and ethical reasons. Life Cycle Assessment provides a well accepted methodology for modeling environmental impacts of human activities. This methodology relies on the decomposition of a studied system into interdependent processes in a step called Life Cycle Inventory. Every process has several environmental impacts and the composition of those processes provides the cumulated environmental impact for the studied human activities. Several organizations provide processes databases containing several thousands of processes with their interdependency links that are used by LCA practitioners to do an LCA study.

Understanding and audit of those databases requires to analyze a huge number of processes and their dependency relations. But those databases can contain thousands of processes linked together. We identified two problems that the experts faces using those databases :

- organize the processes and their dependency relations to improve the comprehensibility ;
- calculate the impacts and, if it is not possible, find why it is not feasible.

In this thesis, we :

- show that there are some semantic similarities between the processes and their dependency relations and propose a new way to model the dependency relations in an inventory database. In our approach, we semantically index the processes using an ontology and we use a multi-layers model of the dependency relations. We also study a declarative approach of this multi-layers approach ;
- propose a method to calculate the environmental impacts of the processes based on linear algebra and graph theory, and we study the conditions of the feasibility of this calculation when we have a cyclic model.

We developed a prototype based on this approach that showed some convincing results on different use cases. We tested our prototype on a case study based on a data set extracted from the National Renewable Energy restricted to the electricity production in the United-States.

PUBLICATIONS

La plupart des idées présentées dans ce document ont fait l'objet des publications suivantes :

CONFÉRENCES INTERNATIONALES

A Semantic Approach to Life Cycle Assessment Applied on Energy Environmental Impact Data Management. B. Bertin, V. Scuturici, J.M. Pinon, E. Risler. Dans Workshop on Energy Data Management (EnDM 2012) in conjunction with EDBT 2012, Berlin, Allemagne. 2012.

Semantic modelling of dependency relations between Life Cycle Analysis processes. B. Bertin, V. Scuturici, J.M. Pinon, E. Risler. Dans ICT-GLOW 2012 in conjunction with DEXA 2012, Vienne, Autriche. 2012.

CarbonDB : a Semantic Life Cycle Inventory Database (demo). B. Bertin, V. Scuturici, J.M. Pinon, E. Risler. Dans Conference on Information and Knowledge Management (CIKM) 2012, Maui, Hawaï. 2012.

CONFÉRENCES NATIONALES

Modélisation sémantique des relations de dépendance entre les processus de l'analyse en cycle de vie. B. Bertin, V. Scuturici, J.M. Pinon, E. Risler. Dans INFORSID 2012, Montpellier, France. 2012.

REVUES NATIONALES

Base de données sémantique d'inventaires en cycle de vie. B. Bertin, V. Scuturici, J.M. Pinon, E. Risler. Dans Ingénierie des systèmes d'information (ISI) 2013, volume 18/2.

REMERCIEMENTS

Je tiens à remercier mes directeurs de thèse, Marian Scuturici et Jean-Marie Pinon, pour la confiance qu'ils m'ont accordé et la qualité des conseils qu'ils m'ont apporté. Je remercie aussi Irène Kryze et Emmanuel Risler qui m'ont permis de réaliser cette thèse dans les meilleures conditions possibles.

Mes remerciements vont aussi à mes deux rapporteurs, Dominique Rieu et Ioan Todinca, pour leur temps et leurs remarques ainsi qu'à mes deux examinateurs, Christine Largeron et Bruno Defude, pour avoir bien voulu participer à la soutenance.

Je remercie mes collègues et amis doctorants avec qui j'ai tant partagé pendant ces dernières années : Arnaud, Brice, Deming, Elise, Pierre-Nicolas, Sabina, Simon et Usman.

Merci aussi à mes collègues de My C-Sense, Valentin et Matthieu, pour la qualité de leur travail qui a grandement simplifié le mien. Je remercie aussi les membres de l'équipe DRIM pour les différents échanges que nous avons eu.

Enfin je remercie mes deux frères Jean-Baptiste et Boris pour leur soutien et nos multiples discussions, Julien avec qui j'ai beaucoup échangé intellectuellement et sur un terrain, ma mère pour ses encouragements et mon père pour son obsession à me pousser toujours plus loin. Et merci Diane d'avoir supporté mes sautes d'humeurs et de m'avoir encouragé pendant tout ce temps.

TABLE DES MATIÈRES

1	INTRODUCTION	1
1.1	Contexte sociétal	1
1.2	Contexte Industriel	2
1.3	Problématiques	3
1.3.1	Modélisation des relations de dépendance	3
1.3.2	Calcul des impacts	5
1.4	Contributions	6
1.5	Structuration de ce document	7
2	ANALYSE ENVIRONNEMENTALE ET BASES DE DONNÉES ENVIRONNEMENTALES	9
2.1	Analyse en cycle de vie	9
2.1.1	Modélisation et méthodologie	10
2.1.2	Limitations et critiques	12
2.1.3	Base de données d'inventaires en cycle de vie	13
2.1.4	Ecoinvent	14
2.2	Autres méthodes d'analyse environnementale	16
2.2.1	Analyse entrée/sortie environnementale	16
2.2.2	Comptabilité carbone	17
2.3	Synthèse	19
3	MODÉLISATION MULTI-NIVEAUX D'UN INVENTAIRE EN CYCLE DE VIE	21
3.1	Exemple d'inventaire	22
3.2	Une méthode basée sur trois graphes	23
3.2.1	Description de la méthode	24
3.2.2	Application à l'exemple d'inventaire	25
3.3	Formalisme	26
3.3.1	Indexation sémantique et ontologie	26
3.3.2	Dimensions de mots-clefs	27
3.3.3	Groupes de processus et de coefficients	28
3.3.4	Mots-clefs communs	29
3.4	Conversion des relations du macro-graphe en relations du graphe détaillé	30
3.4.1	Union de deux ensembles de dimensions	30
3.4.2	Nombre d'appariements	31
3.4.3	Règle de conversion d'une macro-relation	31
3.4.4	Exemple de conversion d'une macro-relation	32
3.4.5	Différents types de macro-relations	34
3.4.6	Algorithmes de conversion d'une macro-relation	36
3.5	Expérimentation sur le passage à l'échelle de l'algorithme de conversion	43
3.6	Synthèse	49
4	ÉTUDE DE CAS DE LA MODÉLISATION MULTI-NIVEAUX	51
4.1	Limites de l'étude : production d'électricité aux Etats-Unis	51

4.2	Macro-graphe et macro-relations	51	
4.2.1	Production d'électricité à partir du charbon	52	
4.2.2	Moyens de transport utilisés pour la production d'électricité à partir de biomasse	53	
4.2.3	Ressources utilisées pour la production d'électricité à partir de pétrole	53	
4.3	Macro-modélisations alternatives	55	
4.4	Synthèse	56	
5	APPROCHES DÉCLARATIVES DE LA MODÉLISATION MULTI-NIVEAUX D'UN INVENTAIRE	57	
5.1	Modélisation multi-niveaux basée sur l'algèbre relationnelle	57	
5.1.1	Multiplication d'un groupe de processus par un groupe de coefficients	58	
5.1.2	Addition de deux groupes de processus	60	
5.1.3	Implémentation en SQL	62	
5.1.4	Expérimentation sur le passage à l'échelle de l'implémentation en SQL	65	
5.1.5	Limites de l'approche basée sur l'algèbre relationnelle	66	
5.2	Approche basée sur les standards du web sémantique	67	
5.2.1	Modélisation OWL de l'ontologie des mots-clefs	68	
5.2.2	Modélisation OWL du graphe détaillé	70	
5.2.3	Requêtes sémantiques et graphe détaillé inverse	73	
5.3	Synthèse	76	
6	CALCUL DES FLUX CUMULÉS	77	
6.1	Calcul matriciel de l'Analyse en Cycle de Vie	77	
6.1.1	Faisabilité du calcul des flux cumulés	78	
6.1.2	Réalisation effective du calcul des flux cumulés	79	
6.2	Optimisation du calcul	82	
6.2.1	Décomposition par bloc de la matrice des flux économiques	82	
6.2.2	Exemple de décomposition	84	
6.2.3	Calcul des flux cumulés dans un système dynamique	84	
6.3	Synthèse	87	
7	PROTOTYPE(S)	89	
7.1	Interfaces	89	
7.1.1	Exploration de la base d'inventaires	89	
7.1.2	Édition de la base d'inventaires	91	
7.2	Fonctionnalités transverses	93	
7.3	Application opérationnelle	94	
7.4	Synthèse	95	
8	BUSINESS PROCESS MODELING	97	
8.1	Gestion des processus métier	97	
8.2	Compréhensibilité des modèles de processus métier	100	
8.3	Abstraction de modèles de processus métier	101	
8.4	Synthèse	108	
9	CONCLUSION	109	
9.1	Contributions	109	
9.2	Perspectives	111	

9.2.1	Génération automatique du macro-graphe	111
9.2.2	Analyse en cycle de vie hybride	111
9.2.3	Calcul des flux cumulés	111

BIBLIOGRAPHIE	113
---------------	-----

TABLE DES FIGURES

FIGURE 1	Sources des impacts environnementaux liés à l'utilisation d'un véhicule. 2
FIGURE 2	Graphe orienté des relations de dépendance entre les processus de transport de marchandises et les processus de production d'électricité. 4
FIGURE 3	Graphe orienté des relations de dépendance entre les processus de production d'électricité aux États-Unis. 5
FIGURE 4	Graphe orienté de la relation de dépendance entre un groupe contenant les processus de transport de marchandises et un groupe contenant les processus de production d'électricité. 6
FIGURE 5	Étapes de l'analyse en cycle de vie. 10
FIGURE 6	L'architecture des applications d'ecoinvent. 15
FIGURE 7	Illustration de la modélisation multi-niveaux d'une base d'inventaires en cycle de vie. 21
FIGURE 8	Les six étapes de la méthode proposée. La première étape correspond à l'ontologie des mots-clefs. La deuxième étape contient deux groupes référençant des processus présentés dans la troisième étape. La quatrième étape contient un macro-graphe. Sa conversion en un graphe détaillé est présentée dans la cinquième étape. La sixième étape contient la matrice des coefficients extraite depuis le graphe détaillé. 25
FIGURE 9	Relation de dépendance entre le groupe des processus de transport de marchandises et le groupe des processus de production d'électricité. 25
FIGURE 10	Représentation graphique d'un groupe de coefficients G_c et de deux groupes de processus G_{p1} et G_{p2} . Le groupe G_{p1} contient des processus de transport de marchandises et est composé d'une dimension. Le groupe G_{p2} contient des processus de production d'électricité et est composé de deux dimensions. Le groupe de coefficients G_c contient les coefficients de dépendance entre les processus de G_{p1} et G_{p2} et est composé de trois dimensions. 26
FIGURE 11	Graphe détaillé de la production d'électricité à partir de pétrole restreint aux processus amonts de transport de marchandises. 26
FIGURE 12	Exemple d'ontologie pour stocker les mots-clefs décrivant des modes de transport. Les nœuds sont des mots-clefs et les étiquettes des arcs sont des prédicats. 27
FIGURE 13	Une macro-relation normale $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées. 35

FIGURE 14	Une macro-relation partielle $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées. Les éléments grisés dans la macro-relation sont ceux qui ne sont pas utilisés dans des relations détaillées. 35
FIGURE 15	Une macro-relation agrégée $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées. 36
FIGURE 16	Une macro-relation projetée $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées. 37
FIGURE 17	Dictionnaire des coefficients indexés par les mots-clefs communs avec le groupe de processus amont. 41
FIGURE 18	Comparaison du temps de calcul pour convertir des macro-relations de taille croissantes entre la version naïve et la version optimisée de l'algorithme de conversion des macro-relations. L'échelle de l'axe des ordonnées est logarithmique. 44
FIGURE 19	Comparaison du temps de calcul pour convertir des macro-relations de taille croissante avec la version optimisée de l'algorithme de conversion pour des relations impliquant des groupes de une, deux, trois et quatre dimensions. 45
FIGURE 20	Comparaison du temps de calcul pour convertir des macro-relations projetées de taille croissante avec la version optimisée de l'algorithme de conversion pour des relations impliquant un groupe de coefficients ou un seul coefficient. L'échelle de l'axe des ordonnées est logarithmique. 46
FIGURE 21	Comparaison du temps de calcul pour convertir des macro-relations agrégées de taille croissante avec la version optimisée de l'algorithme de conversion pour des relations impliquant un groupe de coefficients ou un seul coefficient. L'échelle de l'axe des ordonnées est logarithmique. 46
FIGURE 22	Comparaison du temps de calcul pour convertir des macro-relations de taille croissante impliquant des groupes n'ayant aucun mot-clef en commun entre la version optimisée et la version naïve de l'algorithme de conversion. Ce graphique présente aussi le nombre de relations détaillées créées en fonction de la taille des groupes composant la macro-relation. L'échelle de l'axe des ordonnées pour le temps de calcul est logarithmique. 47
FIGURE 23	Graphe détaillé correspondant au jeu de données extrait de la base de données d'inventaires en cycle de vie du NREL. Les groupes de coefficients ne sont pas représentés pour simplifier la figure. 52
FIGURE 24	Macro-graphe correspondant au jeu de données extrait de la base de données d'inventaires en cycle de vie du NREL. Les coefficients ne sont pas représentés pour simplifier la figure. 53

- FIGURE 25 Conversion en relations détaillées d'une macro-relation $((G_{p1}, G_{p2}), G_c)$. G_{p1} contient les processus de production d'électricité à partir de charbon. G_{p2} contient les processus de production d'électricité pour les sous-régions de l'eGrid. G_c contient les coefficients de dépendance des sous-régions de l'eGrid vis à vis de la production d'électricité à partir du charbon. La notation de l'indexation des éléments est simplifiée : $p_{e,lc}$ correspond au processus $p^{\text{Electricity, Lignite coal}}$. 54
- FIGURE 26 Conversion en relations détaillées d'une macro-relation entre les processus de transport de marchandises et le processus de production d'électricité à partir de biomasse. 55
- FIGURE 27 Conversion en relations détaillées d'une macro-relation entre les processus d'extraction de pétrole et les processus de production d'électricité à partir de pétrole. 56
- FIGURE 28 Alternative de macro-graphe correspondant au jeu de données extrait de la base de données d'inventaires en cycle de vie du NREL. Les coefficients ne sont pas représentés pour simplifier la figure. 56
- FIGURE 29 Modèle Conceptuel de Données de l'implémentation dans un SGBDR. 64
- FIGURE 30 Comparaison du temps de calcul pour convertir des macro-relations de taille croissantes entre la version SQL et les versions procédurales naïve et optimisée de l'algorithme de conversion des macro-relations. L'échelle de l'axe des ordonnées est logarithmique. 66
- FIGURE 31 Comparaison du temps de calcul pour convertir des macro-relations de taille croissantes entre la version SQL et la version procédurale optimisée de l'algorithme de conversion des macro-relations. L'échelle de l'axe des ordonnées est logarithmique. 67
- FIGURE 32 Graphe RDF du vocabulaire de l'ontologie des mots-clefs. 69
- FIGURE 33 Graphe RDF de l'ontologie des mots-clefs correspondant aux modes de transport. 69
- FIGURE 34 Graphe RDF de l'ontologie des mots-clefs correspondant à différents types de camions. Les arcs en pointillés représentent les propriétés inverses inférées par un raisonneur. 70
- FIGURE 35 Graphe RDF des classes OWL des processus et coefficients. 71
- FIGURE 36 Graphe RDF des restrictions sur la classe Element. Les nœuds sans libellé représentent des *blank nodes* RDF. 71
- FIGURE 37 Graphe RDF de la réification des relations détaillées et ses restrictions. 72
- FIGURE 38 Graphe RDF des relations détaillées entre le processus de transport de marchandises par camion et le processus de production d'électricité à partir de charbon. 72

FIGURE 39	Graphe détaillé et sa fermeture transitive, l'arc en pointillé représente la fermeture transitive des deux relations détaillées $((p_a, p_b), c_{ab})$ et $((p_a, p_b), c_{bc})$. 73
FIGURE 40	Graphe détaillé entre trois processus $p_i, i \in \{0, 1, 2\}$. 74
FIGURE 41	Graphe détaillé inverse entre trois processus $p_i, i \in [0, 2]$. 75
FIGURE 42	Graphe RDF de la réification des relations détaillées et ses restrictions. 75
FIGURE 43	Graphe de dépendance entre deux processus p_0 et p_1 . 78
FIGURE 44	Exemple d'une matrice mise sous forme triangulaire supérieure par blocs. 82
FIGURE 45	Graphe détaillé entre six processus p_i avec $0 \leq i \leq 5$. 84
FIGURE 46	Décomposition en composantes fortement connexes du graphe détaillé de la Figure 45. Les nœuds p_{SCC_i} correspondent aux nœuds du graphe G_{SCC} construit à partir de G . Les arcs en pointillés sont les arcs de G_{SCC} . Les nœuds du graphe G_{SCC} sont dans l'ordre topologique. 85
FIGURE 47	Écran d'accueil de l'interface de consultation. Cet écran contient des informations sur le contenu de la base d'inventaires courante, un moteur de recherche et un accès aux différentes bases d'inventaires de l'application. 90
FIGURE 48	Interface d'exploration de la base d'inventaires. 91
FIGURE 49	Éléments de l'interface d'exploration de la base d'inventaires. 92
FIGURE 50	Outil d'édition de l'ontologie des mots-clefs. 92
FIGURE 51	Édition des flux élémentaires d'un processus. 93
FIGURE 52	Historique des actions utilisateurs pour une base d'inventaire. 94
FIGURE 53	Cycle de vie du BPM repris de [50]. 98
FIGURE 54	Exemples de modèles de processus métier avec deux langages de modélisation conceptuels (BPMN et EPC) repris de [92]. 99
FIGURE 55	Distinction entre abstraction par élimination et abstraction par agrégation, repris de [81]. 102
FIGURE 56	Illustration de la méthode d'abstraction de Eshuis et al [25]. 103
FIGURE 57	Illustration de la méthode d'abstraction de Bobrik et al. [8, 9, 10]. 104
FIGURE 58	Illustration de la méthode d'abstraction de Polyvyanyy et al. [69]. 104
FIGURE 59	Règles d'abstraction de Polyvyanyy et al. [70]. 105
FIGURE 60	Illustration de la méthode d'abstraction de processus métier découverts dans les traces d'exécution proposée par Günther et al. dans [37]. 106
FIGURE 61	Illustration de la méthode d'abstraction basée sur la décomposition de processus métier proposée par Polyvyanyy et al. [71]. 107

FIGURE 62 Illustration de la modélisation à deux niveaux d'une base
d'inventaires en cycle de vie. 110

INTRODUCTION

1.1 CONTEXTE SOCIÉTAL

Cette thèse s'inscrit dans le cadre de la mise en place d'outils de mesure des impacts environnementaux des activités humaines. Ces impacts sont multiples et peuvent être, par exemple, les émissions de gaz à effet de serre dans l'atmosphère ou les rejets de matières toxiques dans les sols. Leurs effets ont des répercussions aussi bien sur l'environnement, comme le changement climatique engendré par les émissions de gaz à effet de serre, que sur l'être humain, comme les problèmes sanitaires dus à la pollution des sols par des métaux lourds. Il est primordial de réduire les impacts des activités humaines, notamment les émissions de gaz à effet de serre, en raison de l'imminence et de l'importance des conséquences néfastes qu'ils engendrent. Mais l'étape préalable à la réduction de ces impacts est leur comptabilisation.

C'est l'objectif des méthodes d'analyses environnementales. Ces méthodes reposent sur une modélisation des activités économiques humaines et une évaluation de leurs impacts. Les activités humaines sont interdépendantes : pour que l'une d'entre elles se réalise il faut que d'autres activités se réalisent aussi. La détermination des impacts induits par la réalisation d'une activité requiert de comptabiliser également les impacts issus des activités dont elle dépend. Considérons, par exemple, le cas de l'utilisation d'un véhicule à essence. Son utilisation engendre des émissions de gaz à effet de serre au moment de la combustion de l'essence dans le moteur. Mais pour pouvoir brûler cette essence il a fallu l'extraire, la raffiner, la transporter, etc. Les impacts engendrés par l'utilisation d'une voiture ne sont donc pas déterminés uniquement par son utilisation mais aussi par toutes les étapes en amont et en aval à son utilisation, telles que la production de l'essence, la production du véhicule ainsi que les étapes liées à sa fin de vie. La Figure 1 illustre certaines sources d'impacts environnementaux liées à l'utilisation d'un véhicule.

L'analyse en cycle de vie (ACV) est la méthode d'analyse environnementale la plus utilisée et la plus précise. Une étude basée sur la méthode ACV se décompose en quatre étapes [89]. La première consiste à définir les objectifs et le champ de l'étude. Lors de la deuxième étape, le système étudié est décomposé en plusieurs activités élémentaires pour réaliser un *inventaire en cycle de vie*. Ces activités correspondent à des étapes spécifiques d'un cycle de vie (comme l'extraction de pétrole, son raffinage, son transport, etc.) et peuvent être composées les unes en fonction des autres. Par exemple, la production d'une voiture dépend (la plupart du temps) de la production d'acier. Les troisième et quatrième étapes concernent l'évaluation des impacts et l'interprétation des résultats. Dans la terminologie ACV, les activités d'un inventaire sont nommées

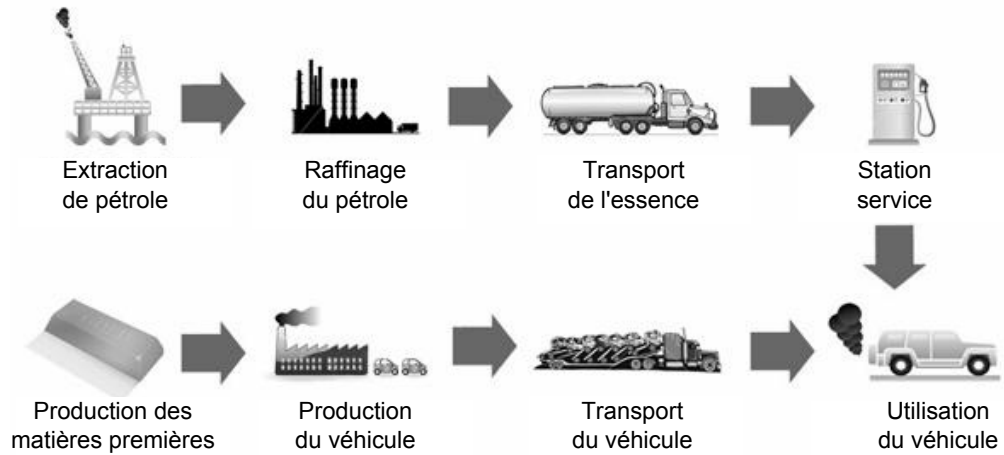


FIGURE 1: Sources des impacts environnementaux liés à l'utilisation d'un véhicule.

processus ou *processus unitaires*¹ [89]. Dans la suite de ce document, nous utiliserons le terme de processus pour nous référer à la notion de processus unitaire de l'ACV.

1.2 CONTEXTE INDUSTRIEL

L'entreprise My C-sense, qui est à l'origine de cette thèse, édite une application web de comptabilité carbone à destination des entreprises ou des collectivités locales. La comptabilité carbone est une méthode d'analyse environnementale qui ne prend en compte que les émissions de gaz à effet de serre. En France, ce type d'analyse est réalisé en appliquant la méthode Bilan Carbone élaborée par l'Agence de l'environnement et de la maîtrise de l'énergie (ADEME). Elle repose sur deux éléments : un outil de calcul (sous la forme d'un classeur Excel) des émissions de gaz à effet de serre d'une entité et un guide des facteurs d'émissions (sous la forme d'un document non structuré). Dans la terminologie de la comptabilité carbone, un facteur d'émission correspond à la quantité de gaz à effet de serre émise dans l'atmosphère pour une activité donnée. Un facteur d'émission peut être par exemple la quantité de gaz à effet de serre émise par une chaudière au fuel pour chauffer un m² de bureau pendant un an.

L'application de cette méthode consiste à réaliser un inventaire des données comptables d'une entreprise ou d'une collectivité locale, telles que la surface des différents locaux de l'entreprise et leurs modes de chauffage (gaz, fuel, électrique). Les données collectées servent ensuite à calculer les émissions de gaz à effet de serre à l'aide des facteurs d'émissions. Par exemple, la surface des bureaux d'une entreprise est multipliée par le facteur d'émissions du chauffage d'un m² de bureau (en fonction du mode de chauffage des locaux) pour obtenir les émissions de gaz à effet de serre liées au chauffage des locaux de

1. A ne pas confondre avec la notion de processus que l'on retrouve dans les systèmes d'information. La notion de processus dans l'ACV est similaire à la notion de tâche ou d'activité des systèmes d'information.

l'entreprise.

La solution éditée par My C-sense est l'équivalent de la méthode Bilan Carbone de l'ADEME. Elle est composée d'un module de collecte de données comptables, d'un module d'analyse et de reporting et d'un module de calcul qui s'appuie sur une base de données de facteurs d'émissions pour déterminer les impacts carbone. L'élaboration de cette base de données de facteurs d'émissions a été le point de départ de cette thèse.

Mais la méthode bilan carbone, et plus généralement la comptabilité carbone, est un sous-ensemble de l'ACV. Les facteurs d'émissions sont similaires aux processus de l'ACV, mais là où l'ACV prend en compte des centaines de rejets de matières dans l'environnement, ils ne prennent en compte que les émissions de gaz à effet de serre. De plus, les facteurs d'émissions de l'ADEME, et des autres sources de facteurs d'émissions que nous avons étudié, sont issus, entre autre, des bases d'inventaires en cycle de vie. Contrairement aux bases d'inventaires en cycle de vie, les sources de données de facteurs d'émissions ne décrivent pas les chaînes de composition des facteurs et fournissent uniquement des valeurs prenant en compte les amonts. C'est pourquoi, pour élargir le champ d'application de notre proposition, nous avons décidé de proposer une méthode qui s'applique aussi bien à l'ACV qu'au bilan carbone.

1.3 PROBLÉMATIQUES

Au cours de cette thèse, nous avons abordé deux problématiques liées aux relations de dépendance entre les processus d'une ACV. La première concerne le grand nombre de relations de dépendance présentes dans un inventaire. La deuxième concerne l'optimisation du calcul des impacts et les conditions portant sa faisabilité.

1.3.1 *Modélisation des relations de dépendance*

Plusieurs organisations proposent des bases de données d'inventaires en cycle de vie [28, 30, 64] qui sont utilisées pour réaliser l'analyse en cycle de vie d'un produit ou d'un service. La méthode usuelle pour modéliser les interactions entre les processus d'un inventaire consiste à énumérer une à une les relations de dépendance entre chaque processus. L'énumération des relations de dépendance est une activité fastidieuse et peut être source d'erreurs. Il existe pourtant des similarités sémantiques entre les processus et leurs relations, tels que les processus de production d'électricité à partir du charbon (à partir de différents types de charbon : lignite, bitumineux, etc.) qui sont tous dépendants des processus de transport. De plus, ces bases de données pouvant contenir plusieurs milliers de processus, il est difficile d'avoir une vue d'ensemble des relations de dépendance, à moins de réaliser une étude détaillée de toute la base. Et les proximités sémantiques entre les relations de dépendance sont difficiles à appréhender quand les relations sont décrites individuellement.

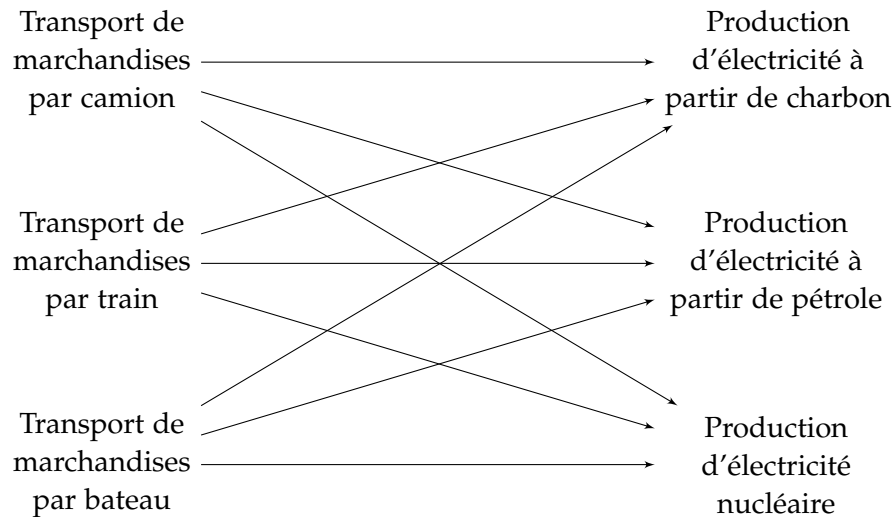


FIGURE 2: Graphe orienté des relations de dépendance entre les processus de transport de marchandises et les processus de production d'électricité.

Nous avons donc cherché à :

- **réduire la quantité de données sources pour décrire les relations de dépendance entre les processus d'un inventaire ;**
- **simplifier la complexité visuelle des relations de dépendance d'un inventaire en cycle de vie.**

Considérons, par exemple, le cas de la production d'électricité à partir d'énergie fossile aux États-Unis. On distingue plusieurs types de centrales électriques dépendant du type de combustible utilisé, tel que le charbon, les combustibles nucléaires et le pétrole. Cette production d'électricité requiert de transporter ces matières fossiles². Nous avons donc des relations de dépendance entre les processus de transport de marchandises et les processus de production d'électricité. En considérant n types d'énergies fossiles et m modes de transport de marchandises, nous avons donc, potentiellement, $n \times m$ relations de dépendance à spécifier. Ce qui, dans la méthode usuelle, implique de préciser m relations de dépendance pour les n types de production d'électricité différents.

De plus, un utilisateur confronté à ces relations de dépendance a seulement une vision locale pour chaque processus. Pour obtenir une vision d'ensemble de ces relations de dépendance, nous pouvons les représenter dans un graphe orienté dans lequel les noeuds correspondent à des processus, les arcs à des relations de dépendance entre des processus et les pondérations des arcs aux coefficients de dépendances entre les processus. La Figure 2 contient le graphe correspondant aux relations de dépendance entre les processus de transport de marchandises et les processus de production d'électricité (nous ne pondérons pas les arcs pour le rendre plus lisible). Sur cet exemple, la représentation dans un graphe orienté permet d'obtenir une vision d'ensemble compréhensible, mais dans le cas d'un inventaire contenant un trop grand nombre de

2. Même si les impacts du transport des matières fossiles sont négligeables face à ceux de la phase de production d'électricité, cet exemple nous permet d'illustrer la problématique et notre contribution.

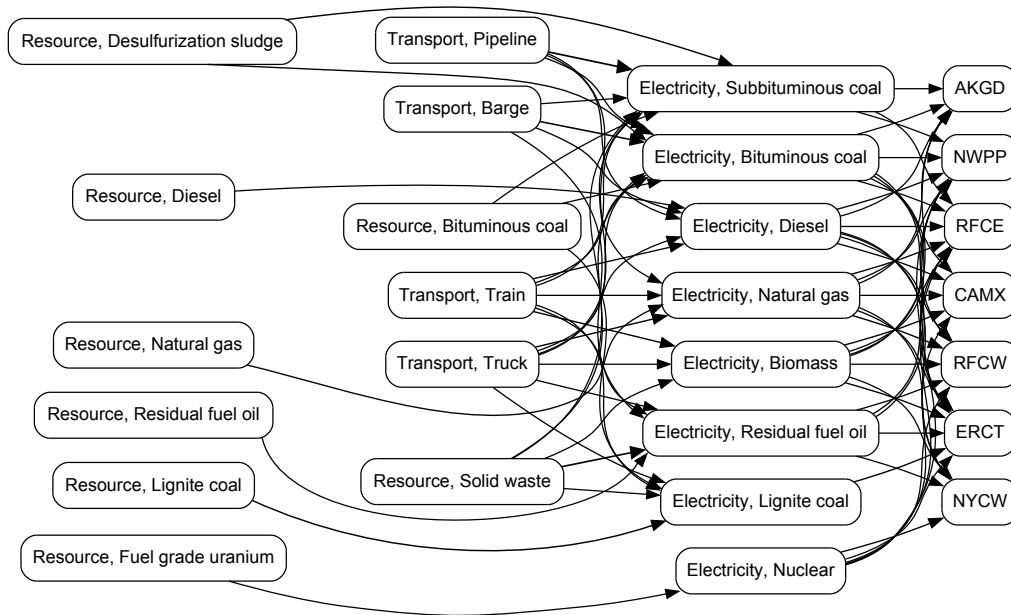


FIGURE 3: Graphe orienté des relations de dépendance entre les processus de production d'électricité aux États-Unis.

noeuds, cette solution n'est pas adéquate. Ce point est illustré dans la Figure 3 qui contient un extrait des relations de dépendance entre les processus de la base d'inventaires américaine maintenue par le National Renewable Energy Laboratory (NREL).

1.3.2 Calcul des impacts

La deuxième problématique que nous avons choisie d'aborder est liée au calcul des impacts des processus et se résume par les deux points suivants :

- **calculer les impacts d'un inventaire le plus rapidement possible ;**
- **détecter les erreurs de modélisation rendant le modèle des relations de dépendance non-calculable.**

Le calcul des impacts des processus d'une base d'inventaires se ramène à un problème d'algèbre linéaire classique : la résolution d'un système d'équation. Mais toutes les modélisations possibles des relations de dépendance ne sont pas forcément calculables, ce qui est le cas si une chaîne de processus fait apparaître un cycle divergent ; en d'autres termes, si on modélise un système qui consomme plus que ce qu'il produit. Dans le cadre d'une application web, le calcul des impacts doit être le plus performant possible, d'autant plus qu'il peut être relativement fréquent si l'on souhaite offrir aux utilisateurs les résultats de chaque modification qu'ils apportent au modèle. Sur ce point, l'orientation de notre travail sur l'ACV plutôt que sur la comptabilité carbone renforce cette contrainte de temps de traitement. Dans les deux méthodes, les données sont

entachées d'incertitude. Mais la caractérisation des incertitudes de l'ACV est beaucoup plus précise que celle de la comptabilité carbone et nécessite d'effectuer une simulation de Monte-Carlo pour calculer les impacts et caractériser leurs incertitudes.

1.4 CONTRIBUTIONS

Notre principale contribution est **une nouvelle méthode pour modéliser une base de données d'inventaires reposant sur la sémantique des processus**. Pour utiliser cette sémantique nous indexons les processus avec des concepts d'une ontologie. Cette indexation nous sert, entre autres, à regrouper sémantiquement les processus. Ces regroupements sont ensuite utilisés pour créer des relations de dépendance entre plusieurs processus au lieu de créer des relations de dépendance entre des processus individuels. Avec cette modélisation nous répondons aux problématiques que nous avons identifiées portant sur les relations de dépendance : nous offrons une vue d'ensemble des données et structurons/facilitons la gestion des relations de dépendance. Cette vue d'ensemble permet une meilleure compréhension d'un modèle des relations de dépendance entre les processus d'un inventaire.

Notre approche est basée sur l'existence de deux niveaux de détails : un premier niveau contenant des relations de dépendance entre des groupes de processus et un deuxième niveau contenant des relations de dépendance entre des processus individuels. Le premier niveau est un graphe orienté pondéré dont les nœuds sont des groupes de processus, les arcs sont des relations de dépendance entre des groupes de processus et les pondérations des arcs sont des groupes de coefficients. Le deuxième niveau correspond au graphe orienté des relations de dépendance entre des processus tel qu'illustré dans la Figure 2. En reprenant l'exemple de cette figure, et en regroupant les processus dont la sémantique est proche, nous pouvons réduire le nombre de relations de dépendance à spécifier et proposer une version simplifiée de ces relations. La Figure 4 illustre la version simplifiée de ce graphe de dépendance entre les processus de transport et les processus de production d'électricité.



FIGURE 4: Graphe orienté de la relation de dépendance entre un groupe contenant les processus de transport de marchandises et un groupe contenant les processus de production d'électricité.

La modélisation se fait alors en exprimant des relations de dépendance entre des groupes de processus. Ces relations doivent ensuite être traduites en relations de dépendance entre des processus individuels afin de calculer les impacts des processus. Nous avons étudié en détail une approche procédurale de l'étape de traduction des relations entre des groupes ainsi qu'une version déclarative basée sur l'algèbre relationnelle. Pour mettre à profit l'indexation sémantique des processus individuels, nous proposons d'analyser les relations

de dépendance d'un processus en utilisant les outils du web sémantique.

Pour valider l'intérêt de cette approche, nous avons développé **deux prototypes de bases de données d'impacts carbone**. Ces prototypes ont amené au développement d'**une application opérationnelle** utilisée par My C-sense reprenant, en partie, l'approche proposée.

Sur le calcul des impacts, nous avons réalisé un état des lieux des méthodes généralement utilisées pour calculer les impacts d'un inventaire, des algorithmes d'algèbre linéaire les plus performants pour réaliser ce calcul et nous avons étudié les conditions de calculabilité d'un inventaire. Nous proposons aussi **une méthode d'optimisation du calcul des impacts suite à un changement unitaire dans une base d'inventaires**, tel que la modification d'un coefficient de dépendance ou l'ajout d'un processus.

1.5 STRUCTURATION DE CE DOCUMENT

Le deuxième chapitre de cette thèse contient une brève étude de trois méthodes d'analyses environnementales : l'analyse en cycle de vie, l'analyse entrée/sortie environnementale et la comptabilité carbone. Nous présentons leurs particularités, expliquons leurs limites et présentons les principales sources de données disponibles. Le troisième chapitre présente la modélisation multi-niveaux d'un inventaire en cycle de vie, les algorithmes utilisés pour la traduction des relations du premier niveau et une étude sur leur passage à l'échelle. Le quatrième chapitre contient une étude de cas de l'application de notre méthode sur des données issues de la base d'inventaires du NREL. Le cinquième chapitre regroupe l'approche déclarative basée sur l'algèbre relationnelle de notre proposition et l'étude sur l'utilisation des outils du web sémantique pour analyser les chaînes de dépendance de processus. Le sixième chapitre contient un état de l'art des méthodes de calcul des impacts des processus et notre proposition sur le calcul dans un environnement dynamique. Le sixième chapitre est consacré aux prototypes et à l'application opérationnelle. La question de la complexité visuelle des relations de dépendances d'un inventaire en cycle de vie se rapproche de la notion de compréhensibilité d'un modèle de processus métier. Nous présentons donc dans le huitième chapitre une synthèse bibliographique des travaux connexes à notre approche multi-niveaux portant sur les problématiques de compréhensibilité et d'abstraction de modèles de processus métier. Le dernier chapitre contient une synthèse de ce document et présente les différentes pistes de recherche pour des travaux futurs.

ANALYSE ENVIRONNEMENTALE ET BASES DE DONNÉES ENVIRONNEMENTALES

L'analyse environnementale ou, plus précisément, l'analyse environnementale des impacts des activités humaines, peut se réaliser en utilisant différentes approches. Le contexte industriel de cette thèse porte sur la méthode Bilan Carbone. Mais les données de cette méthode étant issues de la méthode d'analyse en cycle de vie, et pour élargir le champ d'application de notre proposition, nous avons choisi de nous intéresser à l'analyse en cycle de vie. Nous présentons dans ce chapitre les particularités de ces deux méthodes mais aussi de la méthode d'analyse entrée/sortie environnementale qui apporte une solution à l'une des limitations des deux autres.

Il y a de fortes similitudes entre ces trois approches : elles reposent toutes sur une modélisation des différentes composantes de l'objet de l'étude et sur une estimation de leurs impacts. Ces méthodes diffèrent au niveau du type de sujets étudiés, des impacts environnementaux pris en compte et de la modélisation utilisée. Pour chacune de ces approches, nous présentons des bases de données (ou des sources de données) utilisées par les experts. L'analyse en cycle de vie étant la méthode la plus élaborée pour réaliser des études environnementales détaillées, nous la présentons plus en détail.

2.1 ANALYSE EN CYCLE DE VIE

L'analyse en cycle de vie (ACV) est une méthode utilisée pour déterminer "*l'impact environnemental d'un produit ou d'un service pour toutes les étapes de son cycle de vie - depuis l'extraction des ressources, en passant par la production de matières premières, des pièces du produit lui-même, et de son usage jusqu'à ce qu'il soit détruit, réutilisé ou recyclé*" [36]. Cette méthode permet de tracer les impacts environnementaux systématiques liés à un produit ou service spécifique. Les choix que nous sommes amenés à faire en achetant des produits ou en les concevant peuvent nous permettre de modifier substantiellement nos impacts et d'obtenir une chaîne de production durable.

L'ACV permet d'évaluer les impacts environnementaux dans une chaîne de production impliquant plusieurs acteurs. Les responsabilités portées par chaque acteur peuvent alors être identifiées. Selon l'objectif de l'étude, les impacts de la production d'électricité à partir de charbon peuvent, par exemple, être attribués aux consommateurs de cet électricité plutôt qu'à son producteur qui selon lui ne fait que répondre à une demande. Malgré cette part de subjectivité, cette méthode permet d'avoir un état des lieux à un instant donné des différents impacts environnementaux générés par un produit ou un service.

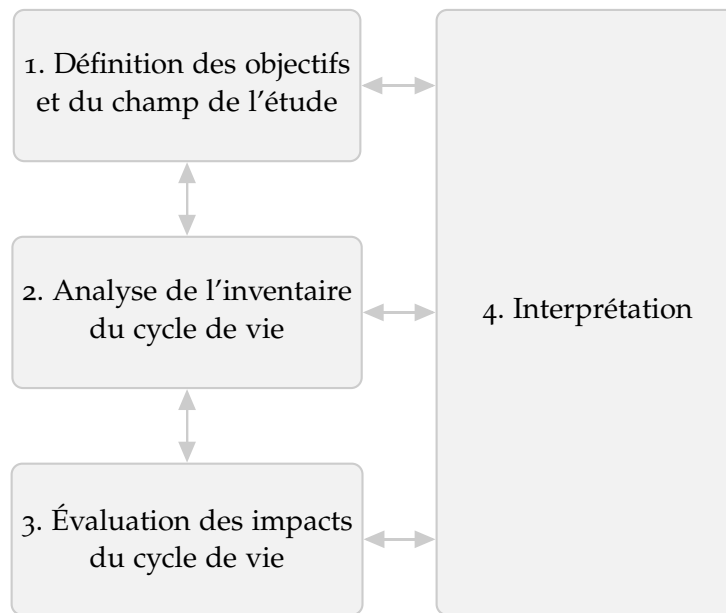


FIGURE 5: Étapes de l'analyse en cycle de vie.

2.1.1 Modélisation et méthodologie

L'ACV a été standardisé dans la série de normes ISO 14000¹. L'analyse en cycle de vie se décompose en quatre étapes interdépendantes telles que présentées dans la Figure 5.

DÉFINITION DES OBJECTIFS ET DU CHAMP DE L'ÉTUDE Cette étape sert à définir quelle est l'utilité de l'étude, son audience et ses motivations. Les objectifs peuvent être l'analyse des points faibles dans la production ou l'optimisation de processus industriels, l'accompagnement à l'analyse du développement de nouveaux matériaux, l'optimisation du cycle de vie d'un produit, etc. Il faut ensuite définir le champ de l'étude en précisant quel est l'objet de l'étude ainsi que ses frontières. Par exemple, pour la production d'électricité, une étude pourrait porter sur l'analyse de la production d'un kilowatt d'électricité sur tout le territoire français en prenant en compte toutes les étapes depuis l'extraction jusqu'à la livraison au compteur du consommateur. Mais on pourrait aussi vouloir analyser les impacts liés au fonctionnement d'une centrale électrique au charbon en Allemagne pour toute sa durée de vie, sans prendre en compte l'étape de distribution de l'électricité.

Ces précisions permettent de définir, pour l'objet de l'étude :

- sa fonction (par exemple : produire de l'électricité ou faire fonctionner une centrale) ;
- son unité fonctionnelle (produire 1 kWh d'électricité ou utiliser une centrale pendant toute sa durée de vie) ;

1. La série de normes ISO 14000 regroupe l'ensemble des normes qui concernent le management environnemental. La norme ISO 14040 [88] définit les principes de l'ACV et la norme ISO 14044 [89] décrit en détail la méthodologie de l'ACV.

- ses paramètres clefs (les pertes en ligne lors du transport d'électricité ou la durée de vie d'une centrale à charbon);
- son périmètre (prendre en compte le transport de l'électricité pour la consommation d'un kWh et prendre en compte la phase de construction et de démantèlement pour l'utilisation d'une centrale).

ANALYSE DE L'INVENTAIRE DU CYCLE DE VIE La deuxième étape consiste à réaliser un inventaire de tous les flux de matières émises et extraites de l'environnement. Dans la terminologie ACV, ces flux sont nommés flux élémentaires. Au cours de cette étape, le système étudié est décomposé en processus interdépendants correspondant aux différentes étapes du cycle de vie de l'objet de l'étude : conception, production, utilisation, recyclage, etc.² Ces processus sont liés entre eux par des flux économiques qui correspondent à des échanges de matière, d'énergie, de services, etc. Ces processus dépendent aussi de processus externes, comme la phase d'extraction de charbon pour la production d'électricité. Pour réaliser une nouvelle analyse en cycle de vie, les experts utilisent des bases de données de processus (aussi nommées bases de données d'inventaires en cycle de vie) pour éviter d'avoir à analyser de nouveau des systèmes déjà étudiés. Les processus d'un inventaire en cycle de vie ou, plus généralement, les processus d'une base d'inventaires, font donc partie d'un réseau de dépendances. La comptabilisation des flux depuis et vers l'environnement à prendre en compte lors de l'avènement d'un processus nécessite donc de prendre en compte ses propres flux élémentaires et les flux élémentaires des processus dont il dépend. Par exemple, pour produire un kWh d'électricité à partir de charbon il faut prendre en compte les émissions de gaz à effet de serre émis lors de la combustion du charbon mais aussi les émissions de gaz à effet de serre émis lors de l'extraction du charbon (et les émissions de gaz à effet de serre des processus amonts de l'extraction, etc.). Les flux élémentaires d'un processus et les flux de ses prédécesseurs forment l'ensemble des flux cumulés d'un processus³.

Un point délicat est lié à l'allocation des flux élémentaires des processus générant plus d'un produit ou service, comme la co-génération de chaleur lors de la production d'électricité. La solution la plus simple consiste à dupliquer le processus en question autant de fois qu'il y a de co-produits et à pondérer les flux économiques en fonction d'une règle d'allocation qui peut être basée sur la valeur économique des co-produits.

ÉVALUATION DES IMPACTS DU CYCLE DE VIE Les flux cumulés obtenus en sortie d'un inventaire en cycle de vie pour un processus sont ensuite agrégés afin d'obtenir des indicateurs d'impacts environnementaux. Cette étape permet de réduire la quantité d'information à prendre en compte pour, par exemple, comparer différents produits ou services : plutôt que d'analyser un grand nombre de flux élémentaires, on ne considère plus que quelques indi-

2. Chacune de ces étapes peut être découpée plus finement pour accroître la précision de l'étude.

3. Le chapitre **CALCUL DES FLUX CUMULÉS** aborde en détail le calcul des flux cumulés des processus d'un inventaire.

cateurs. Les indicateurs d'impacts sont, par exemple, l'acidification des cours d'eaux, le changement climatique ou encore les impacts toxicologiques. L'impact sur le changement climatique est probablement le plus connu : chaque émission de gaz à effet de serre est convertie en équivalent CO₂ en fonction du potentiel de réchauffement global (PRG) du gaz émis. Considérons un processus pour lequel nous avons calculé un flux cumulé de 10 g de dioxyde de carbone (CO₂) et un flux cumulé de 10 g de méthane (CH₄). Sachant que le PRG du CH₄ sur une période de 20 ans est de 72 fois le PRG du CO₂, nous pouvons déterminer un indicateur de ce processus pour le changement climatique qui vaut 82 g équivalent CO₂ sur 20 ans.

Il est aussi possible de calculer un indicateur unique en fonction de tous ces indicateurs pour faciliter la communication sur les effets environnementaux d'un produit ou d'un service. Un tel indicateur est plus subjectif car son calcul consiste à pondérer et agréger les précédents indicateurs. Ceci revient à accorder plus d'importance à certains types de flux élémentaires au détriment d'autres flux⁴.

INTERPRÉTATION Cette étape consiste à analyser les résultats de l'évaluation des impacts, à en expliquer les limites et à proposer des recommandations. Afin de proposer des axes d'amélioration, il faut déterminer quels sont les processus qui engendrent le plus d'impacts. Et pour statuer sur la justesse des résultats il est nécessaire de contrôler leur complétude, la qualité des données utilisées et l'incertitude des résultats. Cette incertitude peut être liée à des simplifications sur la précision des flux élémentaires ou des flux économiques. Par exemple, dans l'analyse de la production d'un kWh d'électricité à partir de charbon, on peut considérer uniquement les valeurs moyennes des flux élémentaires d'un processus d'extraction de charbon pour toute l'Allemagne, au lieu de considérer l'extraction sur un site précis.

2.1.2 Limitations et critiques

La taille d'un inventaire en cycle de vie modélisant les interactions entre un grand nombre de processus se révèle problématique. Plus particulièrement, les bases de données d'inventaires contenant plusieurs milliers de processus et des centaines de milliers de relations de dépendance entre ces processus, sont difficiles à comprendre et à analyser. Ce grand nombre d'éléments les rend aussi difficiles à éditer et à maintenir. Ce problème sera de plus en plus présent car l'un des objectifs des bases d'inventaires est d'être le plus exhaustif et le plus précis possible. Notre proposition sur la modélisation sémantique multi-niveaux des bases de données d'inventaires vise à apporter une solution à ce problème.

L'ACV comporte une part de subjectivité quand au choix des frontières du système étudié. Une étude peut, par exemple, ne pas prendre en compte les

4. Cette critique est aussi valable pour le calcul des autres indicateurs, mais dans une moindre mesure car les experts sont parvenus à un consensus sur une douzaine d'indicateurs objectifs.

processus de production de matière première qui n'ont pas lieu dans la même zone géographique que le processus étudié. La phase d'évaluation des impacts comporte elle aussi une part de subjectivité. Le cas le plus flagrant tient à la détermination d'un indicateur unique pour lequel il n'existe pas une méthode unique, ce qui peut donc, par exemple, favoriser un produit lors d'une étude comparative. Conformément aux obligations de la norme ISO 14040, tous ces choix doivent être présentés dans les conclusions de l'étude. Pour éviter toute source d'interprétation concernant les frontières du système, cette norme précise que le système étudié doit être modélisé de telle façon que ses limites soient uniquement des flux élémentaires (en amont, ce seront donc des "consommations" de ressources).

L'une des plus grandes limitations de la méthodologie ACV tient à la linéarité de son modèle et à la non prise en compte de la temporalité. Des travaux récents s'intéressent au problème de la temporalité en ayant recours à une modélisation multi-agent pour étudier les évolutions possibles d'une modélisation : les agents sont des acteurs économiques qui ont recours à d'autres acteurs économiques et aux ressources fournies par les processus statiques d'une base d'inventaires [20, 19]. Ces travaux permettent de faire évoluer le réseau de dépendances économiques entre les différents processus d'un inventaire. Mais cette critique est aussi valable pour les autres méthodes d'analyse environnementale qui, comme l'ACV, permettent de réaliser un état des lieux à un instant donné.

2.1.3 Base de données d'inventaires en cycle de vie

Certaines bases d'inventaires contiennent des processus pour un secteur économique précis, telles que la base d'inventaires des produits alimentaires produits et consommés au Danemark [61]. D'autres contiennent des processus couvrant plusieurs secteurs économiques mais elles sont restreintes à une zone géographique donnée⁵. Deux de ces bases ont retenu notre attention : la base maintenue par le *National Renewable Energy Laboratory* (NREL) [64] et celle maintenue par la Commission Européenne [26]. La base du NREL se limite au territoire américain et la base européenne au territoire de l'Union Européenne. Nous utilisons les données de la base d'inventaires du NREL pour illustrer certains points de nos propositions. Elle contient plus de 600 processus interdépendants. Le secteur de la production énergétique est le plus détaillé, notamment pour la production d'électricité qui sera le sujet de l'étude de cas que nous présentons dans le chapitre **ÉTUDE DE CAS DE LA MODÉLISATION MULTI-NIVEAUX**.

D'autres bases d'inventaires contiennent des processus couvrant plusieurs secteurs économiques et plusieurs régions géographiques. *Ecoinvent* [28, 29, 27] est la base la plus importante, aussi bien en termes de volumétrie qu'en termes d'usage. Elle contient plus de 4000 processus couvrant le monde entier et prend

5. Elles peuvent contenir des processus extérieurs à cette zone géographique si certains processus de la zone sont dépendants de processus externes ; comme la production de pétrole dans des zones géographiques qui n'ont pas de ressources pétrolières.

en compte environ 400 impacts environnementaux différents. Cette base est aussi la plus structurée et possède l'architecture la plus élaborée, c'est pourquoi nous détaillons son fonctionnement dans la section suivante.

2.1.4 *Ecoinvent*

La base Ecoinvent est maintenue par le centre ecoinvent qui regroupe plusieurs organismes : l'École Polytechnique Fédérale de Lausanne (EPFL), l'institut de technologie de Zurich (ETH Zurich), l'institut Paul Scherrer (PSI), le laboratoire fédéral sur la recherche des matériaux (Empa) et l'institut de recherche agronomique Agroscope. Chaque membre du centre ecoinvent est chargé de la maintenance d'un sous-ensemble de la base. L'accès aux données de cette base est payant⁶ et est disponible dans la plupart des outils d'analyse en cycle de vie. Ecoinvent est aussi à l'origine du format d'échange de données d'inventaires EcoSpold basé sur XML.

L'objectif d'ecoinvent est d'offrir un référentiel qualitatif de processus génériques pour réaliser des analyses en cycles de vie. Ce référentiel couvre les domaines de l'énergie et de la bio-énergie, du transport, de la gestion de déchets, de la construction, de la chimie, des détergents, du papier, de l'agriculture, de l'électronique et de l'ingénierie mécanique.

La philosophie d'ecoinvent est de fournir des données de base sans agrégation et de façon transparente. Chaque activité humaine est donc découpée en plusieurs processus élémentaires⁷.

MODÉLISATION La base ecoinvent contient plusieurs types d'éléments :

- des processus unitaires : des processus qui ne produisent qu'un seul produit/service. Exemple : la production d'une quantité d'acier ;
- des processus à sortie multiples : des processus qui produisent un produit/service principal et des co-produits/services. Exemple : la production d'électricité qui peut avoir comme co-produit la production de chaleur ;
- des flux élémentaires : les rejets dans l'environnement, comme le CO₂ émis ;
- des méthodes de calcul d'impacts : correspondent à l'agrégation des flux élémentaires pour obtenir un impact environnemental, tel que le PRG.

Les processus et les flux élémentaires contenus dans ecoinvent sont classés dans une taxonomie à deux niveaux. Le premier niveau correspond à des secteurs économiques et le deuxième niveau à des ensembles d'activités de ces secteurs économiques. Tous les éléments sont aussi associés à une unité de mesure.

Chaque élément est aussi associé à une zone géographique, la plupart du temps étant un pays mais pouvant aussi être plus large (comme l'union eu-

6. Un accès gratuit existe mais ne permet de consulter qu'un extrait du contenu de la base.

7. Il existe cependant quelques processus agrégés, notamment si aucune étude ne fournit d'informations précises sur la décomposition ou si la décomposition est confidentielle.

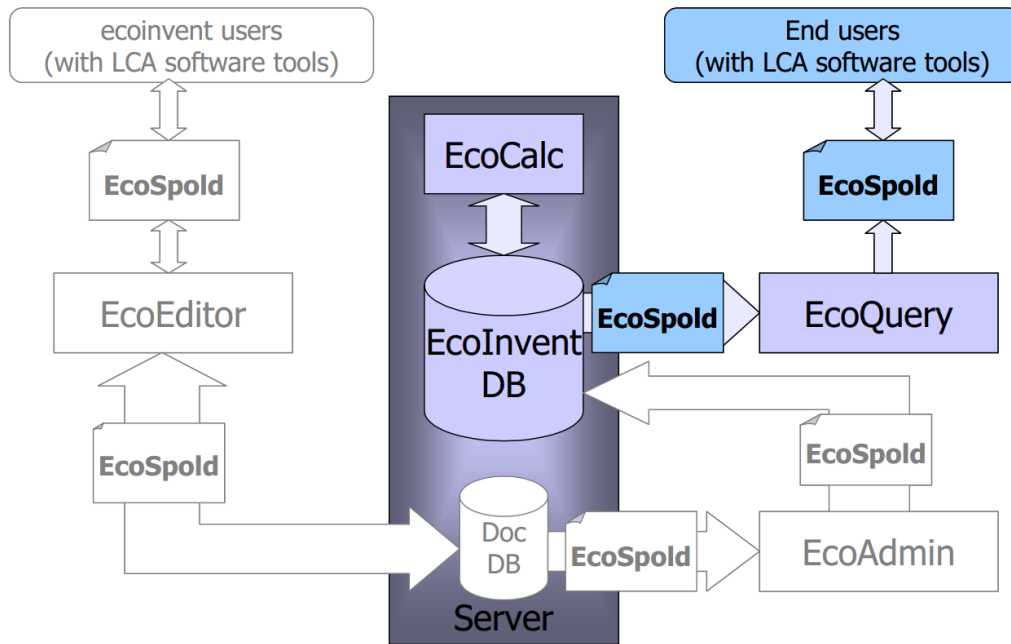


FIGURE 6: L'architecture des applications d'ecoinvent.

ropéenne). Il peut donc y avoir plusieurs processus correspondant à la même activité humaine mais dont la composition et les flux élémentaires sont différents selon la région du monde où ils sont réalisés.

Les données d'ecoinvent sont, pour la plupart, restreintes à la zone géographique européenne. Certains processus hors de ces zones géographiques sont aussi présents s'ils ont des impacts non négligeables et s'ils rentrent en compte dans la composition de processus de la zone géographique européenne. D'autres processus sont associés à une zone géographique très large, ou n'ont pas de zone géographique associée, car ils restent inchangés quel que soit la zone géographique.

ARCHITECTURE Ecoinvent est une base de données supervisées : chaque modification proposée par un éditeur doit être validée par les administrateurs. Le centre ecoinvent livre une nouvelle version régulièrement⁸ pour que les dernières modifications soient utilisées par les experts du domaine. L'écosystème d'applications d'ecoinvent, illustré dans la figure 6, comporte :

- EcoinventDB : une base de données contenant les processus, les flux élémentaires et les méthodes de calcul d'impacts ;
- EcoQuery : une interface web qui permet de parcourir et d'interroger le contenu de la base et d'exporter les données au format EcoSpold ;
- EcoEditor : un éditeur de fichiers EcoSpold ;
- EcoAdmin : un outil d'administration de la base de données ;
- EcoCalc : une application de calcul des impacts.

Tous les flux de données entre ces sous-systèmes se font à l'aide du format EcoSpold.

8. La dernière version accessible au public est la 3.0 qui date de la fin de l'année 2012.

Le workflow d'édition de la base est le suivant : un éditeur propose un nouvel élément ou une modification sur un élément existant, cette proposition est revue par deux autres éditeurs et, si elle est acceptée, elle sera intégrée par un administrateur. Les flux cumulés et les impacts sont ensuite recalculés. Une nouvelle version de la base de données est mise à disposition par les administrateurs de la base regroupant un ensemble de modifications. L'édition des processus peut se faire à l'aide de l'outil EcoEditor ou d'un add-on Excel qui permet de convertir un fichier Excel contenant les mêmes données que le format EcoSpold en un fichier XML au format EcoSpold.

LE FORMAT ECOSPOLD Écrit en XML, il permet d'exporter les données d'un inventaire. Les données présentes dans ce format sont de deux types : les métadonnées sur le processus (références, données administratives, etc.) et les données relatives aux flux (flux élémentaires, flux économiques, règles d'allocation)⁹. Ce format est aussi utilisé pour exporter les données d'autres bases d'inventaires comme celle maintenue par le NREL.

GESTION DES CO-PRODUITS Les processus à sorties multiples d'ecoinvent correspondent à des processus à co-produits et possèdent des règles d'allocation spécifiques vers les produits qui peuvent en découler (tels que la production d'électricité et la production de chaleur pour une centrale électrique). Ces processus, une fois importés dans la base, sont convertis en processus unitaires pour permettre le calcul des inventaires cumulés. Les processus à sorties multiples n'existent donc plus au moment du calcul mais apparaissent encore lors de la consultation de la base.

Les processus d'ecoinvent peuvent être caractérisés comme processus d'infrastructure. Cette distinction permet de ne pas tenir compte des impacts environnementaux qui peuvent être imputés aux infrastructures nécessaires pour la production d'un produit ou d'un service. Mais comme il est souvent difficile de déterminer si un processus fait partie de l'infrastructure ou non¹⁰, les processus d'infrastructure sont, pour l'instant, toujours pris en compte dans les calculs.

2.2 AUTRES MÉTHODES D'ANALYSE ENVIRONNEMENTALE

2.2.1 *Analyse entrée/sortie environnementale*

La qualité d'une étude environnementale est fortement dépendante de la qualité de la (ou des) base(s) de données d'inventaires utilisée(s). Ces bases fournissent un modèle des activités humaines qui peut être incomplet ou peu précis. Pour maximiser la complétude, il existe des bases d'impacts environnementaux basées sur l'approche entrée-sortie de Leontieff appliquée à l'analyse environnementale [52, 42]. L'analyse d'Entrée/Sortie (E/S) permet de décrire les flux

9. Il existe deux autres formats d'échange de données d'ACV : ILCD et ISO14048 [87]. Mais pour la partie concernant l'inventaire, ils sont sensiblement identiques.

10. Dans le cas d'une exploitation pétrolière, le puits de pétrole peut être considéré comme faisant partie de l'infrastructure ou non.

économiques existants entre différents secteurs économiques. Elle permet donc de définir des chaînes de dépendances entre ces secteurs. De même que pour les processus de l'ACV, on peut attribuer à chaque secteur un ensemble de flux vers l'environnement. Il est donc possible de calculer l'ensemble des flux impliqués par les activités d'un secteur en prenant en compte les flux de ses prédécesseurs. Comparée à l'ACV, cette méthode permet de couvrir toutes les activités économiques humaines mais ne fournit pas de résultat aussi précis.

Comme pour l'ACV, certaines bases sont restreintes à une ou plusieurs zones géographiques et couvrent un ou plusieurs secteurs économiques. La base la plus conséquente que nous ayons trouvée est la base EIO-LCA maintenue par l'université Carnegie Mellon [13]. Elle est construite, en partie, à partir des données d'E/S inter-secteurs du bureau d'analyse économique du département du commerce américain. Leur travail a consisté à associer à chaque secteur des émissions de gaz à effet de serre issues d'autres sources, telles que des études scientifiques ou que des données issues de bases d'inventaires.

Ces données sont accessibles au travers d'une interface web permettant de visualiser les dépendances d'un secteur économique avec tous les autres secteurs ou de visualiser la contribution de chaque secteur sur la totalité de ses émissions de gaz à effet de serre. L'interface permet de requêter la base à l'aide d'une classification à deux niveaux sur les secteurs économiques : par secteur économique et par sous-secteur, tel que le secteur agricole et le sous-secteur de la culture de céréales. Cette base étant dépendante de l'activité économique globale, il faut préciser pour quelle quantité d'activité économique on souhaite obtenir les émissions de gaz à effet de serre pour un secteur. On peut, par exemple, obtenir les émissions de gaz à effet de serre correspondant à une activité économique d'un million de dollars du sous-secteur de production de céréales. Nous présentons un exemple des données issues de cette base dans le tableau 1. Il contient les dix premiers secteurs (en termes d'importance de contribution) contribuant aux émissions de gaz à effet de serre de la production de céréales.

2.2.2 Comptabilité carbone

Contrairement aux deux méthodes présentées jusqu'ici, la comptabilité carbone ne prend en compte que les émissions de gaz à effet de serre et s'applique généralement à des sujets d'études peu précis, tels que le fonctionnement d'une entreprise ou l'ensemble de l'activité économique d'un territoire sur une période donnée. On parle d'analyse mono-critères en opposition avec les analyses multi-critères que sont l'ACV et l'analyse E/S environnementale. Cette méthode permet de répondre simplement à la nécessité de comptabiliser les émissions de gaz à effet de serre des acteurs économiques en vue de leur réduction. Mais la simplicité apportée par l'aspect mono-critère correspond aussi à sa principale faiblesse ; un effort de réduction des impacts ne prenant en compte que les gaz à effet de serre peut engendrer une augmentation des émissions

SECTOR	T EQ. CO ₂
<i>Total for all sectors</i>	4470
Grain farming	3360
Fertilizer Manufacturing	373
Power generation and supply	278
Oil and gas extraction	97
Petroleum refineries	63.1
Cattle ranching and farming	36.6
Truck transportation	33.8
All other crop farming	19.8
Pipeline transportation	17.7
Other basic organic chemical manufacturing	17.3

TABLE 1: Émissions de gaz à effet de serre du sous-secteur de la culture de céréales extraites depuis la base de données d'entrée/sortie EIO-LCA

vers d'autres compartiments de l'environnement.

Comme pour l'analyse en cycle de vie, il existe des bases de données permettant de réaliser une comptabilité carbone. Elles couvrent généralement plus d'activités économiques que des bases d'inventaires en cycle de vie car les données qu'elles contiennent sont moins détaillées. La plupart des données proviennent d'autres bases telles que des bases d'inventaires ou des bases d'E/S environnementales. L'objectif est d'offrir une couverture maximum au prix d'une précision plus faible, en utilisant, par exemple, des données issues de sources globales plutôt que des données précises issues d'analyses en cycle de vie. Mais ces bases ne fournissent pas le détail de la composition des activités qu'elles contiennent, nous n'avons donc que l'équivalent des flux cumulés de l'ACV sans les flux élémentaires et les flux économiques. Cette approche, bien que suffisante pour réaliser une comptabilité carbone, n'apporte pas la transparence nécessaire pour assurer la fiabilité des données contenues dans ces bases.

L'Agence De l'Environnement et de la Maîtrise de l'Énergie (ADEME) développe depuis 2002 la méthode Bilan Carbone [1]. C'est un outil de comptabilité carbone pour des entités opérant sur le territoire français. Cette méthode repose sur deux éléments : un outil de comptabilité sous la forme d'un classeur Excel et un guide des facteurs d'émissions¹¹. Le guide des facteurs d'émissions est un document non-structuré regroupant à la fois les données comptables des

11. Dans la terminologie Bilan Carbone, un facteur d'émission s'apparente à un processus de l'analyse en cycle de vie.

	ACV	E/S ENV.	C. CARBONE
<i>Type de données</i>	Processus	E/S	Hybride
<i>Précision</i>	Forte	Faible	Moyenne
<i>Complétude</i>	Faible	Forte	Forte
<i>Impacts</i>	Multi-critères	Multi-critères	Mono-critère
<i>Usage</i>	Produit	Activité	Entité

TABLE 2: Comparaison de trois méthodes d'analyse environnementale

émissions de gaz à effet de serre pour les activités les plus courantes, ainsi que des informations sur leur provenance.

D'autres organismes fournissent des facteurs d'émissions couvrant d'autres territoires. En Angleterre, le Department for Environment, Food and Rural Affairs fournit un ensemble de facteurs d'émissions similaires à ceux de la méthode Bilan Carbone pour le territoire anglais. A un niveau géographique global, le Groupement Intergouvernemental d'Experts sur les évolutions du Climat (GIEC)¹², édite l'Emission Factor Database (EFDB) qui regroupe des facteurs d'émissions pour plusieurs zones géographiques.

2.3 SYNTHÈSE

Le tableau 2 regroupe les trois approches présentées ainsi que leurs principaux points de comparaison. Bien qu'il soit difficile d'obtenir une base de données couvrant toutes les activités humaines (complétude), l'ACV est la méthode qui fournit les données les plus détaillées (précision). L'analyse E/S environnementale ne peut pas atteindre le niveau de précision de l'ACV car les données sources utilisées, à savoir les données économiques par secteur, sont trop agrégées. Cette méthode permet de maximiser la complétude au prix d'une faible précision. Depuis une dizaine d'années, certains groupes de recherche tentent de perfectionner une approche hybride cumulant à la fois l'ACV et l'analyse E/S environnementale proposée dans les années 1970 [83, 84]. Cette approche consiste soit à intégrer les données des bases d'inventaires de l'ACV dans les bases d'E/S environnementales, soit l'inverse. Elle permet d'augmenter la complétude des bases d'inventaires mais se heurte à un problème de taille : la double comptabilisation des impacts, elle n'est donc pas encore utilisée par les experts du domaine.

12. Le GIEC est à l'origine du rapport international sur le changement climatique.

MODÉLISATION MULTI-NIVEAUX D'UN INVENTAIRE EN CYCLE DE VIE

Dans ce chapitre nous présentons notre proposition pour répondre aux problématiques liées à la compréhensibilité des bases d'inventaires et à la réduction de la quantité d'informations utilisées pour décrire des relations de dépendance entre des processus. La méthode que nous proposons repose sur l'existence de deux niveaux de détails pour modéliser les relations de dépendance. Le premier niveau contient des relations de dépendance entre des ensembles de processus sémantiquement proches. Le deuxième niveau contient, comme dans une base de données d'inventaires classique, des relations entre des processus individuels. Toute relation entre des ensembles de processus du premier niveau est convertie en relations entre des processus dans le deuxième niveau. La Figure 7 illustre ces deux niveaux et leurs relations.

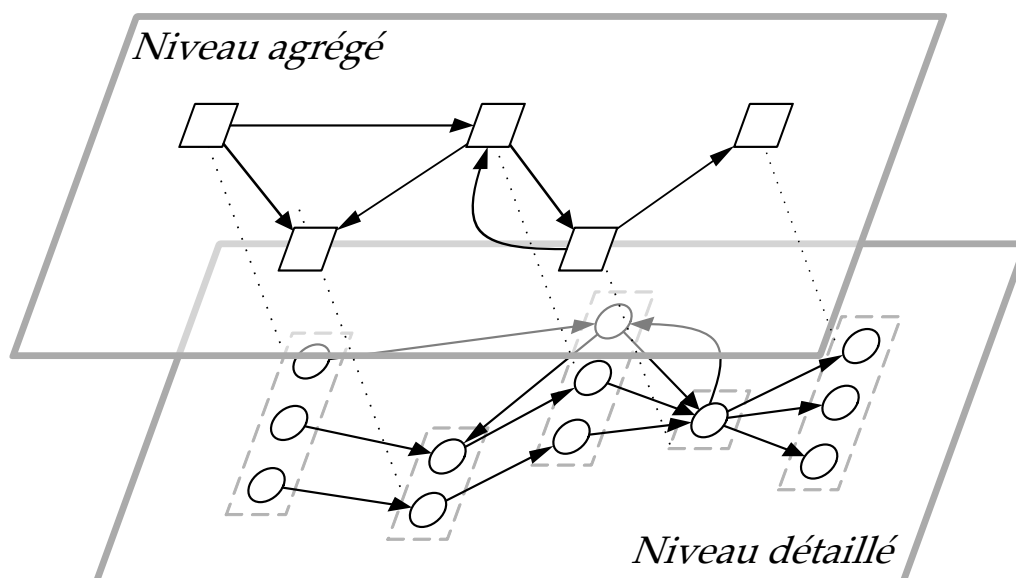


FIGURE 7: Illustration de la modélisation multi-niveaux d'une base d'inventaires en cycle de vie.

Dans la première section, nous présentons un exemple simple d'inventaire en cycle de vie qui nous servira à illustrer certaines notions tout au long de ce chapitre. Dans la deuxième section, nous présentons la méthode de gestion d'une base d'inventaires utilisant cette modélisation multi-niveaux ainsi qu'un exemple pour l'illustrer. Dans la troisième section, nous formalisons les notions de notre approche : l'ontologie des mots-clefs, le graphe détaillé, le macro-graphe et les groupes. Dans la quatrième section, nous présentons la méthode de conversion des relations du macro-graphe en relations du graphe détaillé. Dans la dernière section, nous présentons une étude sur le passage à l'échelle d'une implémentation procédurale de la méthode de conversion.

3.1 EXEMPLE D'INVENTAIRE

Les impacts environnementaux d'une activité humaine sont déterminés en réalisant un inventaire de ses *flux élémentaires* [36]. Un flux élémentaire correspond à une quantité de matière rejetée vers l'environnement, telle qu'une quantité de CO₂ émise dans l'atmosphère ou une quantité de produit polluant rejetée dans un cours d'eau. Dans la modélisation ACV, les activités humaines sont décomposées en processus interdépendants auxquels sont associés un ou plusieurs flux élémentaires. L'avènement d'un processus nécessite de prendre en compte à la fois ses flux élémentaires et les flux élémentaires de ses prédécesseurs. On parle alors des *flux cumulés* d'un processus. Les relations de dépendance entre les processus sont pondérées par des *coefficients* de dépendance. Les flux cumulés d'un processus peuvent donc être exprimés comme une composition linéaire des flux cumulés d'autres processus.

Soit p un processus et $F_c(p)$ l'ensemble de ses flux cumulés. On note p_0, \dots, p_n les prédécesseurs de p (aussi appelés *processus amonts* de p), c_0, \dots, c_n les coefficients de dépendance entre les processus amonts et p . On note $F_c(p_0), \dots, F_c(p_n)$ les flux cumulés des processus amonts et $F_e(p)$ les flux élémentaires de p . On a alors :

$$F_c(p) = F_e(p) + \sum_{i=0}^n (F_c(p_i) * c_i) \quad (1)$$

Considérons le cas de la production d'un kWh d'électricité. Comme il existe plusieurs types de centrales, nous définissons un processus correspondant à la production d'électricité pour chaque type de centrale, tel que les centrales thermiques au charbon et au pétrole. Ces deux processus sont associés à des flux élémentaires, tels que les quantités de dioxyde de carbone et de méthane émises dans l'atmosphère. On note le processus de production d'électricité à partir de charbon p_{ec} et le processus de production d'électricité à partir de pétrole p_{ep} . On représente les flux élémentaires d'un processus par un vecteur dans lequel la première composante correspond à la quantité de CO₂ émise et la deuxième à la quantité de CH₄ émise. Ces deux flux élémentaires sont exprimés en kg de gaz émis dans l'atmosphère. On a :

$$F_e(p_{ec}) = \begin{pmatrix} 1,1 \\ 0,001 \end{pmatrix} \quad F_e(p_{ep}) = \begin{pmatrix} 0,9 \\ 0,008 \end{pmatrix} \quad (2)$$

Mais l'avènement de ces deux processus requiert de transporter des ressources. Nous les associons à deux processus de transport de marchandises par train et par camion, eux aussi, associés à des flux élémentaires. On note p_{tt} le processus de transport d'une tonne de marchandises sur un km (noté tkm) par train et p_{tc} le processus de transport de marchandises par camion, lui aussi exprimé en tkm. On a :

$$F_e(p_{tt}) = \begin{pmatrix} 0,05 \\ 0,0005 \end{pmatrix} \quad F_e(p_{tc}) = \begin{pmatrix} 0,1 \\ 0,003 \end{pmatrix} \quad (3)$$

Pour calculer les flux cumulés des processus p_{ec} et p_{ep} , nous utilisons l'équation 1 sous la forme d'un système d'équations avec les flux élémentaires 2 et 3. On pondère les relations de dépendance entre les processus de production d'électricité et les processus de transport avec des coefficients calculés en fonction de la distance moyenne entre le lieu de production de la ressource et le lieu de production d'électricité. On a :

$$\begin{cases} F_c(p_{ec}) = F_e(p_{ec}) + 0,4F_c(p_{tt}) + 0,2F_c(p_{tc}) \\ F_c(p_{ep}) = F_e(p_{ep}) + 0,1F_c(p_{tt}) + 0,3F_c(p_{tc}) \end{cases} \quad (4)$$

Comme nous n'avons pas défini de prédécesseurs pour les processus p_{tt} et p_{tc} , leurs flux cumulés sont égaux à leurs flux élémentaires. La solution du système 4 est donc :

$$F_c(p_{ec}) = \begin{pmatrix} 1,14 \\ 0,0018 \end{pmatrix} \quad F_c(p_{ep}) = \begin{pmatrix} 0,935 \\ 0,00895 \end{pmatrix} \quad (5)$$

La solution du système 4 est triviale. Mais dans le cas d'une modélisation plus complète ou d'une base de données d'inventaires contenant plusieurs activités différentes, nous pouvons obtenir un système d'équations comportant plusieurs milliers d'inconnus. Nous présentons une méthode de calcul des flux cumulés dans le chapitre **CALCUL DES FLUX CUMULÉS**.

3.2 UNE MÉTHODE BASÉE SUR TROIS GRAPHES

Notre approche est basée sur l'existence de deux niveaux de graphes orientés pondérés. Le premier, que l'on nomme *graphe détaillé*, contient les relations de dépendance entre des processus. Soit $G_d(V_d, E_d)$ le graphe détaillé où l'ensemble des nœuds V_d correspond à l'ensemble des processus, où l'ensemble des arcs E_d correspond aux relations de dépendance entre les processus et où l'ensemble des pondérations correspond à l'ensemble des coefficients. Soit p et p_0, \dots, p_n des nœuds de G_d , un arc entre p_i et p signifie que le processus p dépend de p_i .

Le deuxième, que l'on nomme *macro-graphe*, contient des relations entre des ensembles de processus regroupés en fonction de leurs sémantiques. Nous nommons ces relations de dépendance *macro-relations*. Soit $G_M(V_M, E_M)$ le macro-graphe où l'ensemble des nœuds V_M correspond à l'ensemble des groupes de processus, l'ensemble des arcs E_M correspond aux relations de dépendance entre les groupes de processus et l'ensemble des pondérations des arcs correspond à l'ensemble des groupes de coefficients.

Le macro-graphe offre une vue simplifiée des données contenues dans le graphe détaillé et facilite l'expression de nouvelles relations de dépendance entre des processus sémantiquement proches. Cette simplification permet aussi de consolider la modélisation. Par exemple, la modélisation des relations de dépendance entre les processus de transport de marchandises et les processus de

production d'électricité implique un grand nombre de relations de dépendance. Ceci peut induire des erreurs ou des oublis. La modélisation des relations de dépendance entre des groupes, qui sont traduites en relations entre processus automatiquement, réduit le nombre d'informations sources dans le modèle et réduit donc le nombre potentiel d'erreurs de modélisation.

Afin de regrouper sémantiquement les processus, nous avons choisi de les indexer avec un ensemble de mots-clés qui sont stockés dans une ontologie [34, 53] (qui correspond au troisième graphe). Le vocabulaire de cette ontologie est composé de mots-clés et de prédicats pour créer des relations binaires entre ces mots-clés.

A l'aide de cette ontologie, nous pouvons regrouper les processus et les coefficients dans des groupes sémantiques. Un groupe est similaire à une matrice multidimensionnelle dans laquelle chaque dimension est un ensemble de mots-clés. Ces dimensions sont décrites à l'aide d'une requête sur l'ontologie. Nous pouvons créer des relations de dépendance entre des groupes de processus en utilisant des groupes de coefficients, de la même manière que nous créons des relations de dépendance entre des processus et des coefficients individuels. Nous pouvons ensuite convertir ces relations entre des groupes en relations entre des processus individuels et calculer les flux cumulés.

3.2.1 Description de la méthode

La méthode basée sur ces trois graphes est présentée dans la Figure 8. Avec cette méthode, la création d'une base d'inventaires repose sur six étapes :

1. Créer une ontologie de mots-clés ;
2. Créer des groupes de processus en utilisant cette ontologie ;
3. Comme la base de données est vide, il faut préciser les valeurs numériques des flux des processus et les valeurs numériques des coefficients correspondant aux coordonnées dont nous avons besoin ;
4. Créer des macro-relations entre des groupes ;
5. Convertir le macro-graphe en un graphe détaillé ;
6. Calculer le système d'équations linéaires correspondant à la matrice des coefficients extraite du graphe détaillé.

L'existence, éventuelle, de cycles nous empêche de calculer directement les flux cumulés des processus dans le macro-graphe. Nous sommes donc contraints à le convertir en un graphe détaillé et à calculer les flux cumulés avec la matrice des coefficients¹.

Dans une base de données déjà existante, si nous ajoutons des macro-processus et des macro-relations, les processus référencés par les groupes peuvent déjà exister. Dans ce cas, nous enrichissons le réseau de dépendance de ces processus. Tout changement apporté à l'ontologie peut impliquer des changements à

1. La méthode de calcul des flux cumulés est expliquée dans le chapitre [CALCUL DES FLUX CUMULÉS](#)

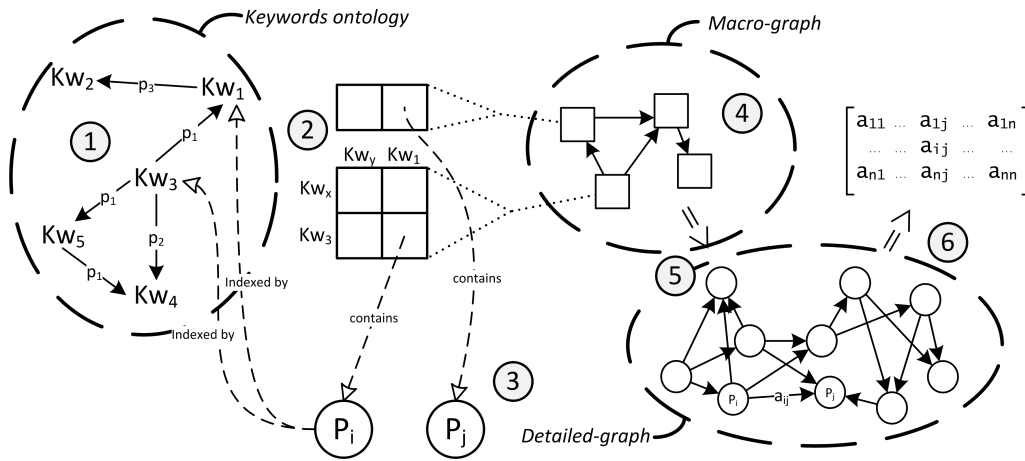


FIGURE 8: Les six étapes de la méthode proposée. La première étape correspond à l’ontologie des mots-clés. La deuxième étape contient deux groupes référençant des processus présentés dans la troisième étape. La quatrième étape contient un macro-graphe. Sa conversion en un graphe détaillé est présentée dans la cinquième étape. La sixième étape contient la matrice des coefficients extraite depuis le graphe détaillé.

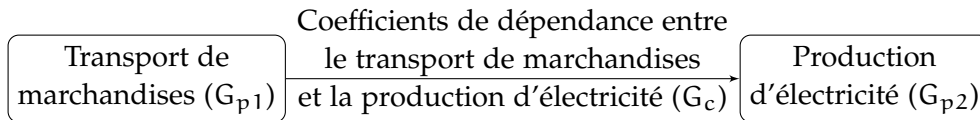


FIGURE 9: Relation de dépendance entre le groupe des processus de transport de marchandises et le groupe des processus de production d’électricité.

la base de données. Il est aussi possible de réutiliser des groupes déjà définis précédemment pour créer de nouvelles relations de dépendance.

3.2.2 Application à l'exemple d'inventaire

Dans notre proposition, au lieu de créer des relations de dépendance entre les quatre processus individuels de notre exemple, nous créons une macro-relation entre un groupe de processus correspondant aux processus de production d’électricité et un groupe de processus correspondant aux processus de transport de marchandises. La Figure 9 présente cette macro-relation.

Les deux groupes de processus et le groupe de coefficients de cette macro-relation sont représentés dans la Figure 10. Le groupe G_{p1} contient les processus de transport de marchandises, le groupe G_{p2} contient les processus de production d’électricité et le groupe G_c contient les coefficients de dépendance entre ces deux groupes. Les termes que l’on retrouve dans les dimensions de ces tableaux correspondent à des mots-clés. La notation p_{Train} utilisée dans cette représentation signifie que ce processus, référencé dans G_{p1} , est indexé par le mot-clé *train*. De la même façon, le coefficient $c_{Train,Électricité,Charbon}$ référencé dans G_c , est indexé par les mots-clés *Train*, *Électricité* et *Charbon*.

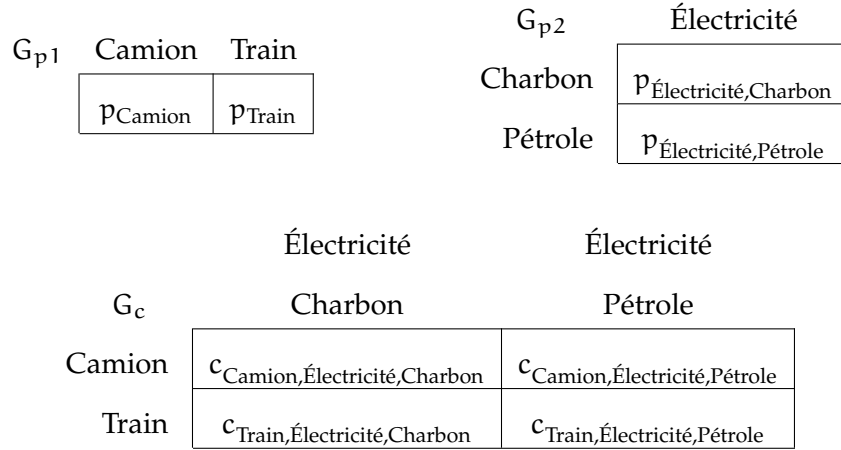


FIGURE 10: Représentation graphique d'un groupe de coefficients G_c et de deux groupes de processus G_{p1} et G_{p2} . Le groupe G_{p1} contient des processus de transport de marchandises et est composé d'une dimension. Le groupe G_{p2} contient des processus de production d'électricité et est composé de deux dimensions. Le groupe de coefficients G_c contient les coefficients de dépendance entre les processus de G_{p1} et G_{p2} et est composé de trois dimensions.

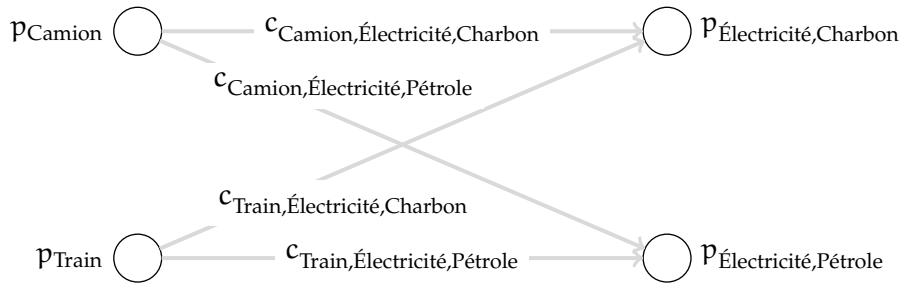


FIGURE 11: Graphe détaillé de la production d'électricité à partir de pétrole restreint aux processus amonts de transport de marchandises.

Cette relation est ensuite convertie en plusieurs relations de dépendance entre les processus individuels contenus dans les groupes de processus en utilisant les coefficients du groupe de coefficients. Le graphe détaillé entre ces quatre processus est présenté dans la Figure 11.

3.3 FORMALISME

3.3.1 Indexation sémantique et ontologie

Dans notre approche, les processus et les coefficients sont indexés et identifiés par des mots-clés qui servent à définir leurs sémantiques. Nous définissons la notion de processus indexé :

Définition 1. Soit k_1, \dots, k_n des mots-clés. Un processus indexé est un couple composé de flux élémentaires et d'un ensemble de mots-clés que l'on note : $p = (F_e(p), K_p)$ où $K_p = \{k_1, \dots, k_n\}$.

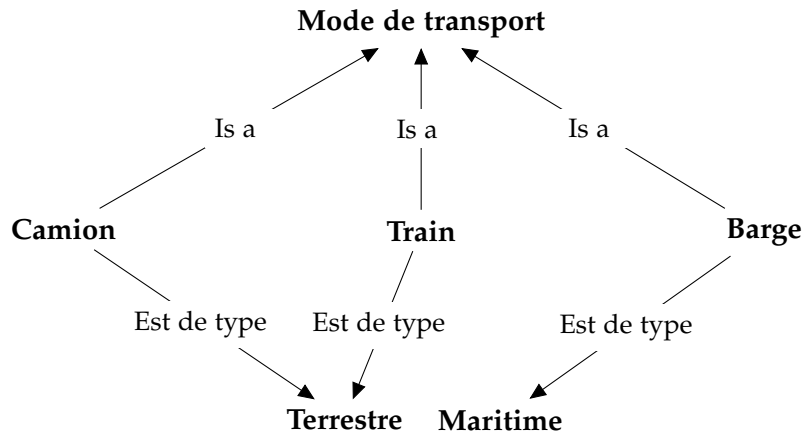


FIGURE 12: Exemple d'ontologie pour stocker les mots-clefs décrivant des modes de transport. Les nœuds sont des mots-clefs et les étiquettes des arcs sont des prédicats.

Contrairement à un processus qui est associé à des flux élémentaires, un coefficient n'est associé qu'à une valeur scalaire que l'on note $V(c)$ pour un coefficient c . De façon analogue à la définition d'un processus indexé, nous définissons la notion de coefficient indexé :

Définition 2. Soit k_1, \dots, k_n des mots-clefs. Un coefficient indexé est un couple composé d'une valeur scalaire et d'un ensemble de mots-clefs que l'on note : $c = (V(c), K_c)$ où $K_c = \{k_1, \dots, k_n\}$.

Par exemple, le processus de transport par camion est identifié par le mots-clef : *Camion*. Il s'écrit donc sous la forme : $p_{\text{Camion}} = (F(p_{\text{Camion}}), \{\text{Camion}\})$. L'ontologie dont nous avons besoin pour décrire les mots-clefs est relativement simple : elle contient des mots-clefs et des prédicats. Les mots-clefs peuvent être reliés entre eux en utilisant des prédicats. Nous imposons une contrainte sur l'indexation : il n'y a qu'un seul processus ou coefficient associé à un ensemble de mots-clefs. Cette contrainte est nécessaire pour la conversion des macro-relations et le déréférencement des éléments d'un groupe.

3.3.2 Dimensions de mots-clefs

Pour exprimer les notions de groupe de processus et de groupe de coefficients nous définissons la notion de *dimension*. Une dimension est un ensemble disjoint de mots-clefs appartenant à l'ontologie des mots-clefs :

Définition 3. Soit S une ontologie de mots-clefs. Une dimension $D = \{k_1, \dots, k_n\}$ est un sous-ensemble de S tel que $\forall i, j \in \{1, \dots, n\}, i \neq j, k_i \in D, k_j \in D, k_i \neq k_j$.

Les groupes de processus et de coefficients possèdent plusieurs dimensions et nous imposons que toutes leurs dimensions soient disjointes deux à deux (*i.e.* : les dimensions d'un groupe n'ont aucun mots-clefs en commun). Nous définissons la notion d'*ensemble de dimensions disjoint* :

Définition 4. Soit (D_0, \dots, D_n) un ensemble de dimensions. Un ensemble de dimensions disjointes est noté $\mathcal{D} = (D_0, \dots, D_n)$ tel que $\forall (i, j) \in \mathbb{N}^2, i \neq j, D_i \cap D_j = \emptyset$.

Dans la suite de ce document, par souci de simplification, nous écrivons "ensemble de dimensions" à la place d'"ensemble de dimensions disjointes".

Deux ensembles de dimensions sont dits *compatibles* si leurs dimensions ne sont en correspondance qu'avec, au plus, une seule dimension de l'autre ensemble. Une dimension est en correspondance avec une autre dimension si leur intersection est non nulle. Nous définissons la notion de compatibilité d'ensemble de dimensions telle que :

Définition 5. Deux éléments \mathcal{D} et \mathcal{D}' de $\mathcal{P}(\mathcal{P}(S))$ sont compatibles si $\forall D \in \mathcal{D}, |\{D' \in \mathcal{D}' \mid D \cap D' \neq \emptyset\}| \leq 1$ et $\forall D' \in \mathcal{D}', |\{D \in \mathcal{D} \mid D \cap D' \neq \emptyset\}| \leq 1$.

Exemple : soit $\mathcal{D}, \mathcal{D}'$ et \mathcal{D}'' trois ensembles de dimensions tels que :

$$\begin{aligned}\mathcal{D} &= \{D_1, D_2\} = \{\{\text{Camion}, \text{Train}\}, \{\text{Diesel}, \text{Essence}\}\} \\ \mathcal{D}' &= \{D_3\} = \{\{\text{Camion}, \text{Avion}\}\} \\ \mathcal{D}'' &= \{D_4\} = \{\{\text{Train}, \text{Diesel}\}\}\end{aligned}$$

Les ensembles \mathcal{D} et \mathcal{D}' sont compatibles, alors que \mathcal{D} et \mathcal{D}'' sont incompatibles car $D_1 \cap D_4 = \{\text{Train}\}$ et $D_2 \cap D_4 = \{\text{Diesel}\}$.

L'ontologie des mots-clefs nous permet de définir dynamiquement les groupes. Une dimension est le résultat d'une requête sur l'ontologie. A partir de l'ontologie présentée dans la Figure 12, nous pouvons exprimer une requête pour créer une dimension contenant tous les modes de transport. Cette requête récupère tous les mots-clefs reliés au mot-clef *Mode de transport* en ne considérant que le prédicat *is a*. Nous pouvons aussi récupérer uniquement les modes de transport terrestres en récupérant l'intersection entre l'ensemble des mots-clefs reliés au mot-clef *Mode de transport* à l'aide du prédicat *is a* et l'ensemble des mots-clefs reliés au mot-clef *Terrestre* à l'aide du prédicat *Est de type*.

Cette définition dynamique des dimensions des groupes apporte un autre avantage à notre approche. Tout changement apporté à l'ontologie des mots-clefs peut déclencher une mise à jour des dimensions précédemment définies, modifiant ainsi les groupes déjà existants. Par exemple, si un mode de transport est ajouté à l'ontologie de la Figure 12 (tel que *voiture*), chaque groupe qui possède une dimension contenant les modes de transport terrestres sera mis à jour.

3.3.3 Groupes de processus et de coefficients

Avec les notions de dimension et d'ensemble de dimensions nous définissons la notion de groupe de processus tel que :

Définition 6. Soit P l'ensemble des processus. On appelle groupe de processus le couple (\mathcal{D}, P_p) où $\mathcal{D} = (D_0, \dots, D_n)$ est un ensemble de dimensions et P_p est une application $P_p : D_0 \times \dots \times D_n \rightarrow P$.

Une autre notation pour exprimer un groupe de processus est basée sur l'énumération de ses processus :

$$G_p = (\mathcal{D}, P_p) = (\{D_1, \dots, D_n\}, \{p_1, \dots, p_n\})$$

$$G_p = (\{D_1, \dots, D_n\}, \{ (F(p_1), K_{p_1}), \dots, (F(p_n), K_{p_n}) \})$$

Similairement, nous définissons la notion de groupe de coefficients telle que :

Définition 7. Soit C l'ensemble des coefficients. On appelle groupe de coefficients le couple (\mathcal{D}, C_c) où $\mathcal{D} = (D_0, \dots, D_n)$ est un ensemble de dimensions et C_c est une application $C_c : D_0 \times \dots \times D_n \rightarrow C$.

Nous pouvons aussi exprimer un groupe de coefficients comme une énumération :

$$G_c = (\mathcal{D}, C_c) = (\{D_1, \dots, D_n\}, \{c_1, \dots, c_n\})$$

$$G_c = (\{D_1, \dots, D_n\}, \{ (V(c_1), K_{c_1}), \dots, (V(c_n), K_{c_n}) \})$$

Le groupe de processus G_{p1} et le groupe de coefficients G_c que nous avons présentés dans la Figure 11 s'écrivent alors :

$$G_{p1} = (\mathcal{D}_{p1}, P_{p1}) = (\{D_{p1_1}\}, P_{p1}) = (\{ \{Camion, Train\} \}, P_{p1})$$

$$G_c = (\mathcal{D}_c, C_c) = (\{D_{c1}, D_{c2}, D_{c3}\}, C_c)$$

$$G_c = (\{ \{Camion, Train\}, \{Électricité\}, \{Charbon, Pétrole\} \}, C_c)$$

Par extension de la définition de la compatibilité de deux ensembles de dimensions, deux groupes de processus sont compatibles si et seulement si leurs ensembles de dimensions sont compatibles. Par exemple, les groupes G_{p1} et G_c sont compatibles car D_{p1_1} a une intersection non nulle uniquement avec D_{c1} . Une macro-relation entre un groupe de processus amont G_{p1} , un groupe de coefficients G_c et un groupe de processus aval G_{p2} est valide si les trois groupes sont compatibles entre eux (G_{p1} doit être compatible avec G_c et G_{p2} , G_c doit être compatible avec G_{p2}).

3.3.4 Mots-clefs communs

Il est fréquent de vouloir exprimer deux groupes avec des dimensions identiques. Par exemple, on peut avoir deux groupes avec une même dimension contenant les mots-clefs correspondant aux différents types de pétrole. Le premier groupe contient des processus correspondant à la phase d'extraction et ceux du deuxième groupe correspondent à la phase de raffinage. Les processus référencés dans ces deux groupes seront donc retrouvés en utilisant la même indexation ; ils référenceront donc les mêmes processus.

Nous introduisons la notion de *mots-clefs communs* qui se comportent comme des dimensions avec un seul mot-clef. Un mot-clef commun sert au déréférencement de tous les éléments contenus dans les cellules d'un groupe. Ces mots-clefs communs sont des entités différentes des dimensions d'un point de vue pratique, mais se comportent exactement comme des dimensions d'un point de

vue théorique. Nous notons l'ensemble des mots-clefs communs d'un groupe \mathcal{C} et, dans la notation d'un groupe, nous les faisons apparaître au même titre que les dimensions. Un groupe de processus est donc noté :

$$G_p = (\mathcal{D}_p, \mathcal{C}_p, P_p) = (\{D_1, \dots, D_n\}, \{k_1, \dots, k_n\}, \{p_1, \dots, p_n\})$$

Nous pouvons aussi intégrer les mots-clefs communs sous la forme de dimensions à un élément dans la liste des dimensions d'un groupe :

$$G_p = (\mathcal{D}_p, P_p) = (\{D_1, \dots, D_n, \{k_1^{\mathcal{C}}\}, \dots, \{k_n^{\mathcal{C}}\}\}, \{p_1, \dots, p_n\})$$

3.4 CONVERSION DES RELATIONS DU MACRO-GRAPHE EN RELATIONS DU GRAPHE DÉTAILLÉ

Pour calculer les flux cumulés, nous devons convertir les arcs du macro-graphe en arcs du graphe détaillé. Puis, nous pouvons extraire la matrice des coefficients² du système d'équations permettant de calculer les flux cumulés.

Une macro-relation correspond à un arc dans le macro-graphe, pondéré par un groupe de coefficients. Plus formellement, on note une telle relation sous la forme : $((G_{\text{amont}}, G_{\text{aval}}), G_{\text{coeff}})$ où G_{amont} et G_{aval} sont des groupes de processus et G_{coeff} est un groupe de coefficients. Cette relation doit être traduite en un ensemble de relations détaillées (*i.e.* : à un ensemble d'arcs dans le graphe détaillé) que l'on note : $\{(p_i, p_j), c_{ij}\}$ où p_i et p_j sont des processus et c_{ij} est un coefficient, tel que $p_i \in G_{\text{amont}}$, $p_j \in G_{\text{aval}}$ et $c_{ij} \in G_{\text{coeff}}$. Seuls les processus et les coefficients des groupes de processus et du groupe de coefficients qui ont une indexation commune seront reliés entre eux dans le graphe détaillé.

Pour réaliser cette traduction nous avons besoin d'introduire deux notions : l'union de deux ensembles de dimensions et le nombre d'appariements entre deux ensembles de dimensions.

3.4.1 Union de deux ensembles de dimensions

L'union de deux ensembles de dimensions correspond à la réunion de trois ensembles :

1. l'intersection des dimensions des deux ensembles qui ont une intersection non nulle (qui ont donc des mots-clefs en commun)
2. toutes les dimensions du premier ensemble de dimensions qui n'ont pas d'intersection non nulle avec les dimensions du deuxième ensemble
3. toutes les dimensions du deuxième ensemble de dimensions qui n'ont pas d'intersection non nulle avec les dimensions du premier ensemble

2. La matrice des coefficients est construite en extrayant les pondérations des arcs du graphe détaillé : un arc entre deux processus p_i et p_j pondéré par un coefficient c correspondra à un coefficient de la matrice des coefficients a_{ij} tel que $a_{ij} = c$.

Définition 8. On note \cup_D l'opérateur d'union de deux ensembles de dimensions tel que, pour deux ensembles de dimensions \mathcal{D}_1 et \mathcal{D}_2 :

$$\begin{aligned} \mathcal{D}_1 \cup_D \mathcal{D}_2 = & \{D_1 \cap D_2 \mid D_1 \in \mathcal{D}_1 \wedge D_2 \in \mathcal{D}_2 \wedge D_1 \cap D_2 \neq \emptyset\} \\ & \cup \{D_1 \mid D_1 \in \mathcal{D}_1 \wedge \forall D_2 \in \mathcal{D}_2, D_1 \cap D_2 = \emptyset\} \\ & \cup \{D_2 \mid D_2 \in \mathcal{D}_2 \wedge \forall D_1 \in \mathcal{D}_1, D_2 \cap D_1 = \emptyset\} \end{aligned}$$

Par exemple, soit \mathcal{D}_1 et \mathcal{D}_2 deux ensembles de dimensions tels que :

$$\begin{aligned} \mathcal{D}_1 = \{D_1, D_2\} &= \{\{a, b, c\}, \{e, f\}\} \\ \mathcal{D}_2 = \{D_3, D_4\} &= \{\{a, b, d\}, \{g, h\}\} \end{aligned}$$

On a :

$$\mathcal{D}_1 \cup_D \mathcal{D}_2 = \{\{a, b\}, \{e, f\}, \{g, h\}\}$$

Car :

$$\begin{aligned} D_1 \cap D_3 &= \{a, b, c\} \cap \{a, b, d\} = \{a, b\} \\ \forall D \in \mathcal{D}_2, D_2 \cap D &= \emptyset \\ \forall D \in \mathcal{D}_1, D_4 \cap D &= \emptyset \end{aligned}$$

3.4.2 Nombre d'appariements

Le nombre d'appariements entre deux ensembles de dimensions correspond au nombre de paires de dimensions de chaque ensemble qui ont une intersection non nulle.

Définition 9. On note $\alpha(\mathcal{D}_1, \mathcal{D}_2)$ le nombre d'appariements entre deux ensembles de dimensions \mathcal{D}_1 et \mathcal{D}_2 tel que :

$$\begin{aligned} \alpha(\mathcal{D}_1, \mathcal{D}_2) = & |\{ (D_1, D_2) \mid D_1 \in \mathcal{D}_1 \\ & \wedge D_2 \in \mathcal{D}_2 \\ & \wedge D_1 \cap D_2 \neq \emptyset \}| \end{aligned}$$

En reprenant les deux ensembles de dimensions \mathcal{D}_1 et \mathcal{D}_2 de l'exemple précédent, on a $\alpha(\mathcal{D}_1, \mathcal{D}_2) = 1$ car seuls D_1 et D_3 ont une intersection non nulle.

3.4.3 Règle de conversion d'une macro-relation

A partir de ces deux notions, nous pouvons définir la règle de conversion d'une macro-relation. Soit G_{p1} et G_{p2} deux groupes de processus, soit G_c un groupe de coefficients et soit $((G_{p1}, G_{p2}), G_c)$ une macro-relation. La règle de conversion de cette macro-relation en relations détaillées est :

$$\begin{aligned} ((G_{p1}, G_{p2}), G_c) \rightarrow & \{((p_1, p_2), c) \mid p_1 \in P_{p1} \wedge p_2 \in P_{p2} \wedge c \in C \\ & \wedge |K_{p1} \cap K_c| = \alpha(\mathcal{D}_1, \mathcal{D}_c) \\ & \wedge |(K_{p1} \cup K_c) \cap K_{p2}| \\ & = \alpha(\mathcal{D}_{p1} \cup_D \mathcal{D}_c, \mathcal{D}_{p2})\} \end{aligned}$$

Pour convertir une macro-relation, on doit donc associer un processus du groupe amont à tous les coefficients du groupe de coefficients et à tous les processus du groupe aval. La règle de conversion peut se découper en deux étapes :

1. la multiplication de tous les processus amonts et des coefficients qui partagent une indexation commune
2. l'association des éléments résultant de l'étape précédente et des processus avals qui partagent une indexation commune

A la fin de la première étape, nous obtenons un ensemble de processus³ *temporaires* que l'on peut ensuite associer à des processus avals.

Étant donné que les trois groupes doivent être compatibles entre eux, si le cardinal de l'intersection des mots-clefs d'un processus amont avec les mots-clefs d'un coefficient est égal au nombre d'appariements $\alpha(\mathcal{D}_1, \mathcal{D}_c)$, cela signifie que les deux éléments partagent une indexation commune et peuvent donc potentiellement faire partie d'une relation dans le graphe détaillé. Comme le groupe amont et le groupe de coefficients sont compatibles, chaque dimension du groupe amont a une intersection non nulle avec, au plus, une seule dimension du groupe de coefficients.

Tous les éléments d'un groupe (de processus ou de coefficients) sont indexés par un mot-clef de chaque dimension. Ils sont donc indexés par autant de mots-clefs qu'il y a de dimensions dans le groupe. L'ensemble des dimensions d'un groupe étant disjoint (il n'y a pas de dimensions qui s'intersectent), le résultat de l'intersection des mot-clefs d'un processus amont avec les mot-clefs d'un coefficient contiendra donc au maximum autant de mots-clefs que le plus petit ensemble de dimensions :

$$\forall p_i \in P_1, c \in C, \max(|K_{p_1}, K_{p_2}|) = \alpha(\mathcal{D}_1, \mathcal{D}_2)$$

3.4.4 Exemple de conversion d'une macro-relation

Pour simplifier la présentation des exemples, on note p_{ab} un processus indexé par les mots-clefs a et b.

Soit $((G_{p1}, G_{p2}), G_c)$ une macro-relation telle que :

$$\begin{aligned} G_{p1} &= (\mathcal{D}_{p1}, P_{p1}) \\ G_c &= (\mathcal{D}_c, P_c) \end{aligned}$$

3. Un processus multiplié par un coefficient donne un processus (ce n'est que la multiplication d'un processus par un scalaire).

avec :

$$\mathcal{D}_{p1} = \{D_{p1_1}, D_{p1_2}, \{x\}\} = \{\{a, b\}, \{c, d\}, \{x\}\}$$

$$P_{p1} = \{p_{acx}, p_{adx}, p_{bcx}, p_{bdx}\}$$

$$\mathcal{D}_c = \{D_{c_1}, D_{c_2}\} = \{\{a, e\}, \{c, d\}\}$$

$$C = \{c_{ac}, p_{ad}, p_{ec}, p_{ed}\}$$

$$\mathcal{D}_{p2} = \{D_{p2_1}, D_{p2_2}, \{y\}\} = \{\{a, b\}, \{c, h\}, \{y\}\}$$

$$P_{p2} = \{p_{acy}, p_{ahy}, p_{bcy}, p_{bhy}\}$$

Pour la première étape, on détermine le nombre d'appariements entre \mathcal{D}_{p1} et \mathcal{D}_c que l'on note α_1 :

$$\alpha(\mathcal{D}_{p1}, \mathcal{D}_c) = \alpha_1 = 2$$

Puis on cherche les paires (processus amont, coefficient) que l'on peut multiplier en testant chaque combinaison. Pour les combinaisons contenant le processus amont p_{ac} on a :

$$\text{card}(K_{p_{acx}} \cap K_{c_{ac}}) = \text{card}(\{a, c, x\} \cap \{a, c\}) = 2 = \alpha_1$$

$$\text{card}(K_{p_{acx}} \cap K_{c_{ad}}) = \text{card}(\{a, c, x\} \cap \{a, d\}) = 1 \neq \alpha_1$$

$$\text{card}(K_{p_{acx}} \cap K_{c_{ec}}) = \text{card}(\{a, c, x\} \cap \{e, c\}) = 1 \neq \alpha_1$$

$$\text{card}(K_{p_{acx}} \cap K_{c_{ed}}) = \text{card}(\{a, c, x\} \cap \{e, d\}) = 0 \neq \alpha_1$$

Donc le processus p_{acx} ne peut s'appareiller qu'avec le coefficient c_{ac} . De la même façon, le processus p_{adx} sera appareillé avec c_{ad} .

Inversement, le processus amont p_{bcx} ne sera appareillé avec aucun coefficient car :

$$\text{card}(K_{p_{bcx}} \cap K_{c_{ac}}) = \text{card}(\{b, c, x\} \cap \{a, c\}) = 1 \neq \alpha_1$$

$$\text{card}(K_{p_{bcx}} \cap K_{c_{ad}}) = \text{card}(\{b, c, x\} \cap \{a, d\}) = 0 \neq \alpha_1$$

$$\text{card}(K_{p_{bcx}} \cap K_{c_{ec}}) = \text{card}(\{b, c, x\} \cap \{e, c\}) = 1 \neq \alpha_1$$

$$\text{card}(K_{p_{bcx}} \cap K_{c_{ed}}) = \text{card}(\{b, c, x\} \cap \{e, d\}) = 0 \neq \alpha_1$$

Il en va de même pour le processus p_{bdx} . Nous obtenons donc les processus temporaires suivants :

$$p_{acx}^t = p_{acx} * c_{ac}$$

$$p_{adx}^t = p_{adx} * c_{ad}$$

L'indexation d'un processus temporaire est l'union des mots-clés du processus amont et du coefficient qui le compose.

Pour la deuxième étape, il faut déterminer l'union des dimensions du groupe de processus amont et du groupe de coefficients :

$$\begin{aligned} \mathcal{D}_{p1} \cup_D \mathcal{D}_c &= \{\{a, b\}, \{c, d\}, \{x\}\} \cup_D \{\{a, e\}, \{c, d\}\} \\ &= \{\{a\}, \{c, d\}, \{x\}\} \end{aligned}$$

Puis le nombre d'appariements entre cet ensemble de dimensions et \mathcal{D}_{p_2} que l'on note α_2 :

$$\begin{aligned}\alpha(\mathcal{D}_{p_1} \cup_D \mathcal{D}_c, \mathcal{D}_{p_2}) &= \alpha(\{\{a\}, \{c, d\}, \{x\}\}, \{\{a, b\}, \{c, h\}, \{y\}\}) \\ &= \alpha_2 = 2\end{aligned}$$

Il faut ensuite essayer d'appareiller les processus temporaires obtenus à la suite de la première étape avec les processus aval. Le processus temporaire p_{acx}^t ne peut s'appareiller qu'avec le processus aval p_{acy} car :

$$\begin{aligned}\text{card}(K_{p_{acx}^t} \cap K_{p_{acy}}) &= \text{card}(\{a, c, x\} \cap \{a, c, y\}) = 2 = \alpha_2 \\ \text{card}(K_{p_{acx}^t} \cap K_{p_{ahy}}) &= \text{card}(\{a, c, x\} \cap \{a, h, y\}) = 1 \neq \alpha_2 \\ \text{card}(K_{p_{acx}^t} \cap K_{p_{bcy}}) &= \text{card}(\{a, c, x\} \cap \{b, c, y\}) = 1 \neq \alpha_2 \\ \text{card}(K_{p_{acx}^t} \cap K_{p_{bhy}}) &= \text{card}(\{a, c, x\} \cap \{b, h, y\}) = 0 \neq \alpha_2\end{aligned}$$

Le processus temporaire p_{adx}^t ne peut s'appareiller avec aucun processus aval car :

$$\begin{aligned}\text{card}(K_{p_{adx}^t} \cap K_{p_{acy}}) &= \text{card}(\{a, d, x\} \cap \{a, c, y\}) = 1 = \alpha_2 \\ \text{card}(K_{p_{adx}^t} \cap K_{p_{ahy}}) &= \text{card}(\{a, d, x\} \cap \{a, h, y\}) = 1 \neq \alpha_2 \\ \text{card}(K_{p_{adx}^t} \cap K_{p_{bcy}}) &= \text{card}(\{a, d, x\} \cap \{b, c, y\}) = 0 \neq \alpha_2 \\ \text{card}(K_{p_{adx}^t} \cap K_{p_{bhy}}) &= \text{card}(\{a, d, x\} \cap \{b, h, y\}) = 0 \neq \alpha_2\end{aligned}$$

Finalement, nous obtenons la relation détaillée $((p_{acx}, p_{acy}), c_{ac})$.

La conversion d'un arc du macro-graphe nécessite d'essayer d'associer chaque processus du groupe amont G_{p_1} avec un coefficient du groupe de coefficients G_c et un processus du groupe aval G_{p_2} . Étant donné que les trois groupes doivent être compatibles entre eux, si le cardinal de l'intersection des mots-clefs d'un processus amont avec les mots-clefs d'un coefficient est égal au nombre d'appariements $\alpha(\mathcal{D}_1, \mathcal{D}_c)$, cela signifie que les deux éléments partagent une indexation commune et peuvent donc potentiellement faire partie d'une relation dans le graphe détaillé. Le résultat de la combinaison de ces deux éléments sera indexé par l'union de leurs mots-clefs. La condition $\text{card}((K_{p_1} \cup K_c) \cap K_{p_2}) = \alpha(\mathcal{D}_1 \cup_D \mathcal{D}_c, \mathcal{D}_2)$ jouera donc le même rôle pour l'association de cette combinaison avec un processus aval que dans le cas d'un processus amont et d'un coefficient. L'union des deux ensembles de dimensions \mathcal{D}_1 et \mathcal{D}_c avec l'opérateur \cup_D est le pendant de l'union des mots-clefs d'un processus amont avec les mots-clefs d'un coefficient.

3.4.5 Différents types de macro-relations

En fonction des dimensions des groupes impliqués dans les macro-relations, nous pouvons classer les macro-relations en quatre types :

MACRO-RELATION NORMALE : les trois groupes ont les mêmes dimensions. C'est le type de relation le plus fréquent. La Figure 13 illustre ce type de macro-relation pour des groupes de processus de deux dimensions.

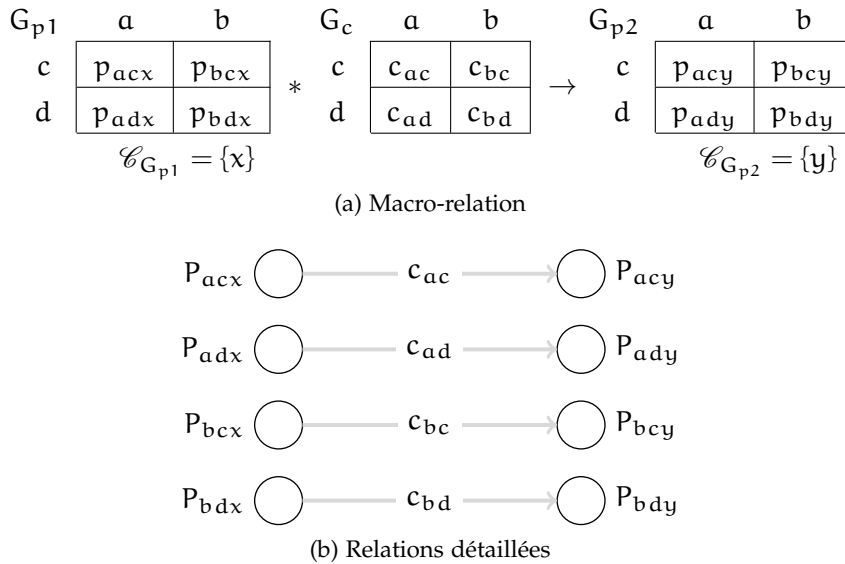


FIGURE 13: Une macro-relation normale $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées.

MACRO-RELATION PARTIELLE : les trois groupes ont le même nombre de dimensions mais au moins une intersection (non nulle) de dimensions appartenant à différents groupes est plus petite que les dimensions intervenant dans cette intersection. Ce type de relation est utile si nous souhaitons utiliser un groupe dans plusieurs relations sans pour autant modéliser des relations détaillées avec chaque processus de ce groupe. La Figure 14 illustre ce type de macro-relation.

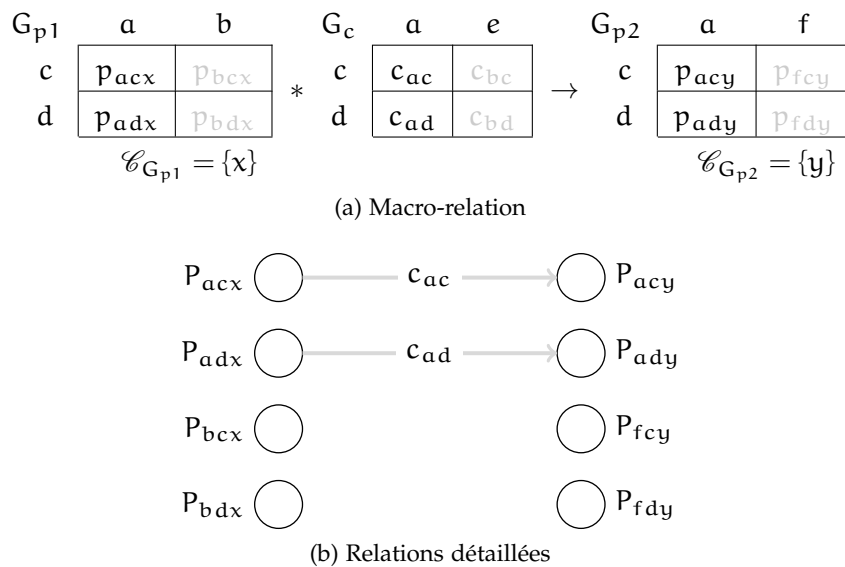


FIGURE 14: Une macro-relation partielle $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées. Les éléments grisés dans la macro-relation sont ceux qui ne sont pas utilisés dans des relations détaillées.

MACRO-RELATION AGRÉGÉE : le groupe aval a moins de dimensions que le résultat de la multiplication du groupe amont par le groupe de coefficients. Si nous souhaitons modéliser des relations entre un processus et tous ses prédécesseurs, nous pouvons créer un groupe les contenant tous et créer une relation entre ce groupe et le processus étudié. La Figure 15 illustre ce type de macro-relation.

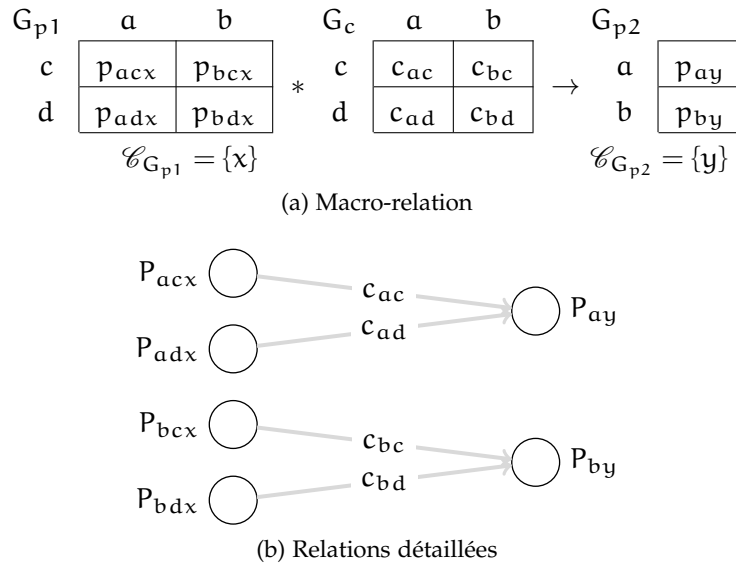


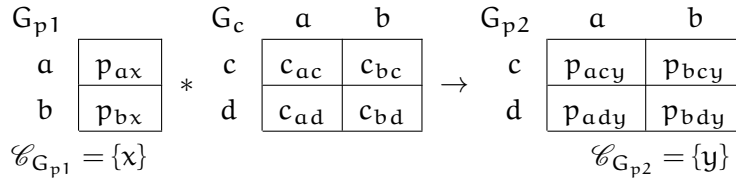
FIGURE 15: Une macro-relation agrégée $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées.

MACRO-RELATION PROJETÉE : le groupe aval a plus de dimensions que le résultat de la multiplication du groupe amont par le groupe de coefficients. Ce type de relation est utile pour factoriser les relations entre un processus et plusieurs processus : les processus avals sont regroupés et nous créons une relation entre le processus amont et ce regroupement (ce type de relation fonctionne aussi si nous avons plus d'un processus amont mais moins de processus avals). La Figure 16 illustre ce type de macro-relation.

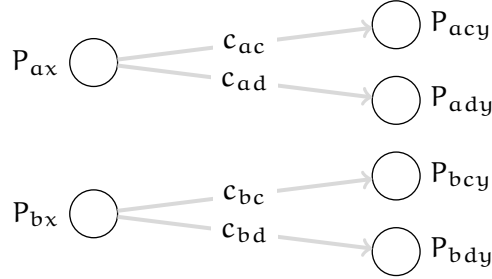
Les différents types de relations peuvent être combinés pour obtenir des relations encore plus expressives telles qu'une relation agrégée-projetée, partielle-agrégée, etc. Le chapitre **ÉTUDE DE CAS DE LA MODÉLISATION MULTI-NIVEAUX** présente des illustrations pour ces types de relations.

3.4.6 Algorithmes de conversion d'une macro-relation

L'algorithme de conversion d'une macro-relation est une application de la règle de conversion présentée dans la section précédente. Comme cette règle, il utilise la notion de nombre d'appariements entre deux ensembles de dimensions et la notion d'union de deux ensembles de dimensions. Les deux premières parties de cette section présentent les versions procédurales des algorithmes correspondant à ces notions, puis nous présentons l'algorithme de



(a) Macro-relation



(b) Relations détaillées

FIGURE 16: Une macro-relation projetée $((G_{p1}, G_{p2}), G_c)$ et sa conversion en relations détaillées.

conversion. Pour chacun de ces algorithmes nous prenons comme point de départ une version naïve de leurs implémentations que nous optimisons.

3.4.6.1 Calcul du nombre d'appariements

Le calcul du nombre d'appariements entre deux ensembles de dimensions \mathcal{D}_1 et \mathcal{D}_2 nécessite de compter le nombre de couples de dimensions dont l'intersection est non nulle. Une version naïve de cet algorithme consiste à tester si chaque dimension de \mathcal{D}_1 a une intersection non nulle avec une dimension de \mathcal{D}_2 en parcourant tous les mots-clefs de toutes les dimensions de \mathcal{D}_1 et de \mathcal{D}_2 .

Ce calcul étant toujours réalisé avec les ensembles de dimensions des groupes d'une macro-relation, \mathcal{D}_1 et \mathcal{D}_2 sont consistants et compatibles entre eux. Nous pouvons donc optimiser cet algorithme de deux manières différentes :

- en stockant tous les mots-clefs \mathcal{D}_2 dans un dictionnaire (ou tableau associatif)⁴ et en testant si l'un des mots-clefs de chaque dimension de \mathcal{D}_1 se retrouve dans ce dictionnaire. Les deux ensembles étant compatibles, si on trouve un mot-clef d'une dimension de \mathcal{D}_2 dans \mathcal{D}_1 on peut arrêter la recherche pour cette dimension et passer à la suivante ;
- comme les dimensions sont consistantes, dès que nous avons déterminé qu'un mot-clef d'une dimension de \mathcal{D}_1 est présent dans l'ensemble de dimensions \mathcal{D}_2 , nous pouvons passer à la dimension suivante de \mathcal{D}_1 .

Dans tous les cas, nous devons parcourir toutes les dimensions de \mathcal{D}_1 et tous les mots-clefs de ces dimensions pour tester si on a une intersection non nulle

4. Chaque mot-clef est identifiable par une adresse unique, telle qu'une URI si on sérialise l'ontologie des mots-clefs en RDF/XML. Nous créons donc un dictionnaire ne contenant qu'un seul élément par adresse. Elle est en fait utilisée pour accélérer le processus de recherche d'un mot-clef dans l'ensemble des mots-clefs d'un ensemble de dimensions.

avec une dimension de \mathcal{D}_2 . Dans le pire des cas, qui correspond à un nombre d'appariements égal à zéro⁵, il est nécessaire de parcourir tous les mots-clefs des dimensions de \mathcal{D}_1 . En considérant que la recherche d'un élément dans un dictionnaire se fait dans le pire des cas⁶ en $O(\log(n))$, dans notre cas on aura $n = \sum_{D \in \mathcal{D}_2} |D|$, la complexité dans le pire des cas est donc $O(m \log(n))$ avec $m = \sum_{D \in \mathcal{D}_1} |D|$. Cet algorithme est présenté dans le listing 1.

Algorithme 1: Calcul du nombre d'appariements

Entrées : $\mathcal{D}_1, \mathcal{D}_2$
\leftarrow table de hachage des mots-clefs de \mathcal{D}_2
nombreAppariements \leftarrow 0
pour chaque dimension $\in \mathcal{D}_1$ **faire**
 pour chaque motClef \in dimension **faire**
 si motClef \in tableHachage **alors**
 nombreAppariements ++
 Passer à la prochaine dimension de \mathcal{D}_1
retourner nombreAppariements

3.4.6.2 Union de deux ensembles de dimensions

De façon similaire, nous pouvons calculer l'union de deux ensembles de dimensions \mathcal{D}_1 et \mathcal{D}_2 en utilisant un dictionnaire pour déterminer si un mot-clef de \mathcal{D}_1 se trouve dans une dimension de \mathcal{D}_2 . Ce dictionnaire, en plus de permettre une recherche rapide, indique quelle est la dimension d'appartenance des mots-clefs qu'il stocke.

Nous parcourons donc les dimensions de \mathcal{D}_1 en construisant l'ensemble de dimensions résultant de cette opération, et :

- si une dimension de \mathcal{D}_1 a une intersection non nulle avec une dimension de \mathcal{D}_2 (c'est-à-dire si un de ses mots-clefs se retrouve dans le dictionnaire), on ajoute l'intersection de ces deux dimensions au résultat et on marque la dimension de \mathcal{D}_2 correspondante comme déjà utilisée ;
- si une dimension de \mathcal{D}_1 n'a aucune intersection non vide avec une dimension de \mathcal{D}_2 , nous l'ajoutons au résultat ;
- nous rajoutons toutes les dimensions de \mathcal{D}_2 non utilisées dans le résultat.

Cet algorithme est présenté dans le listing 2. Nous disposons d'une structure de données permettant de stocker des ensembles de dimensions et une autre pour stocker des dimensions. Ces deux structures possèdent un opérateur += permettant d'ajouter un élément (soit un mot-clef dans une dimension, soit une dimension dans un ensemble de dimensions).

5. S'il n'y a aucune dimension des deux ensembles qui ont une intersection non nulle.

6. Un dictionnaire implémenté en utilisant un arbre équilibré a une complexité pour la recherche dans le pire des cas en $O(\log(n))$, une implémentation utilisant une table de hachage aura une complexité dans le pire des cas en $O(n)$.

Algorithme 2: Calcul de l'union de deux ensembles de dimensions

Entrées : $\mathcal{D}_1, \mathcal{D}_2$
\leftarrow table de hachage des mots-clefs de \mathcal{D}_2
union $\leftarrow \emptyset$ // union est un ensemble de dimensions
pour chaque dimension $\in \mathcal{D}_1$ **faire**
 nouvelleDimension $\leftarrow \emptyset$
 pour chaque motClef \in dimension **faire**
 si motClef \in tableHachage **alors**
 nouvelleDimension += motClef
 marquer dimension comme utilisée
 si nouvelleDimension = \emptyset **alors**
 union += dimension
 sinon
 union += nouvelleDimension
pour chaque dimension $\in \mathcal{D}_2$ *non utilisée* **faire**
 union += dimension
retourner union

Dans tous les cas, toutes les dimensions de \mathcal{D}_1 et tous leurs mots-clefs seront parcourus. Le pire des cas est trouvé quand l'union de \mathcal{D}_1 et \mathcal{D}_2 est égale à l'union de toutes leurs dimensions⁷. Il est alors nécessaire de parcourir en plus toutes les dimensions de \mathcal{D}_2 . La complexité dans le pire des cas est donc $O(n \log(n) + m)$ avec $n = \sum_{D \in \mathcal{D}_1} |D|$ et $m = |\mathcal{D}_2|$. En considérant qu'on a $m \ll n$, la complexité dans le pire des cas est $O(n \log(n))$.

3.4.6.3 Conversion d'une macro-relation

L'algorithme du listing 3 présente une implémentation naïve de la méthode de conversion d'une macro-relation $((G_{p1}, G_{p2}), G_c)$ en un ensemble de relations détaillées. Dans cette version, nous parcourons tous les éléments des trois groupes donc nous testons l'association de $|P_{G_{p1}}| * |P_{G_c}| * |P_{G_{p2}}|$ triplets (contenant un processus amont, un coefficient et un processus aval). Nous disposons d'une structure de données pour stocker des processus et des coefficients qui permet d'accéder aux mots-clefs de leurs indexations *via* l'appel à *element.motClef*. Nous avons aussi une structure de données permettant de stocker les relations détaillées qui possèdent un opérateur += permettant d'ajouter des éléments. Dans tous les cas, la complexité de cet algorithme est de $O(|P_{G_{p1}}| * |P_{G_c}| * |P_{G_{p2}}|)$. En considérant que tous les groupes ont le même nombre d'éléments, on peut approximer cette complexité par $O(n^3)$.

Une première optimisation évidente consiste à tester si l'intersection des mots-clefs d'un couple (processus amont, coefficient) est égal à $\alpha(\mathcal{D}_{p1}, \mathcal{D}_c)$ avant de parcourir tous les processus de G_{p2} . Cette condition, que nous appellerons condition d'appariement, correspond à la première partie de la ligne 4 du listing 3. De cette façon, nous ne testons pas tous les triplets possibles,

7. Aucune des dimensions de \mathcal{D}_1 et \mathcal{D}_2 n'ont de mots-clefs en commun. Autrement dit $\alpha(\mathcal{D}_1, \mathcal{D}_1) = 0$.

Algorithme 3: Version naïve de la conversion d'une macro-relation en un ensemble de relations détaillées

Entrées : G_{p1}, G_{p2}, G_c

- 1 relationsDétaillées $\leftarrow \emptyset$
- 2 **pour chaque** $p_1 \in G_{p1}.processus$ **faire**
- 3 **pour chaque** $c \in G_c.coefficients$ **faire**
- 4 **pour chaque** $p_2 \in G_{p2}.processus$ **faire**
- 5 **si** $p_1.motClef \cap c.motClef = \alpha(\mathcal{D}_1, \mathcal{D}_c)$ **et** $(p_1.motClef \cup$
 $c.motClef) \cap p_2.motClef = \alpha(\mathcal{D}_1 \cup_D \mathcal{D}_c, \mathcal{D}_2)$ **alors**
- 6 relationsDétaillées += $((p_1, p_2), c)$
- 7 **retourner** relationsDétaillées

mais uniquement les triplets dont le couple (processus amont, coefficient) satisfait la condition d'appariement.

Mais nous pouvons aussi complètement nous passer des conditions d'appariement. Il nous suffit de stocker les coefficients et les processus aval dans des tableaux associatifs en fonction des mots-clefs en commun entre les trois groupes.

Nous devons construire deux listes de mots-clefs en commun entre des groupes :

- les mots-clefs en commun entre \mathcal{D}_{p1} et \mathcal{D}_c que nous pouvons calculer lors du calcul de l'union des dimensions du groupe amont et du groupe de coefficients ;
- les mots-clefs en commun entre $\mathcal{D}_{p1} \cup_D \mathcal{D}_c$ et \mathcal{D}_{p2} .

Chaque coefficient est alors indexé en fonction des mots-clefs qu'il a en commun avec le groupe de processus amont. Et chaque processus aval est indexé en fonction des mots-clefs qu'il a en commun avec le résultat de $\mathcal{D}_{p1} \cup_D \mathcal{D}_c$.

Nous construisons l'index des éléments d'un groupe à l'aide d'une fonction de hachage en concaténant les mots-clefs dans l'ordre lexicographique⁸. Donc, pour un processus amont, nous n'avons pas à parcourir tous les coefficients pour trouver ceux qui partagent une indexation commune avec lui, mais nous retrouvons directement (avec un coût dans le pire des cas de $O(\log(n))$) les coefficients à lui associer. Il en va de même pour associer un processus aval au résultat de la multiplication d'un processus amont avec un coefficient.

Pour illustrer la construction du tableau associatif des coefficients, nous considérons une macro-relation impliquant un groupe de processus amont G_{p1} et un groupe de coefficients G_c tels que :

$$G_{p1} = (\mathcal{D}_{p1}, P_{p1})$$

$$G_c = (\mathcal{D}_c, C_c)$$

8. Dans une implémentation utilisant RDF pour l'ontologie, nous pouvons construire les index à l'aide des URI. Chaque URI étant unique, il est possible de les ordonner dans un ordre lexicographique.

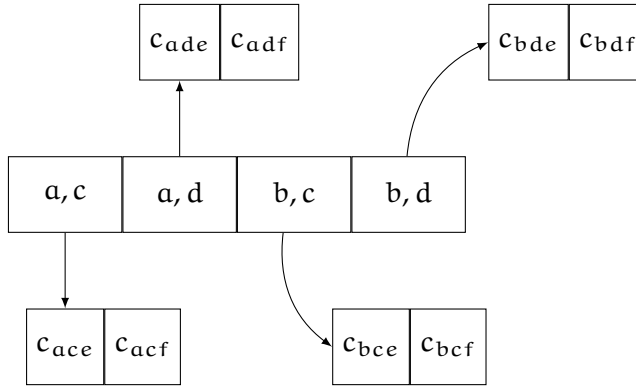


FIGURE 17: Dictionnaire des coefficients indexés par les mots-clefs communs avec le groupe de processus amont.

avec :

$$\mathcal{D}_{p1} = \{D_{p1_1}, D_{p1_2}\} = \{\{a, b\}, \{c, d\}\}$$

$$P_{p1} = \{p_{ac}, p_{ad}, p_{bc}, p_{bd}\}$$

$$\mathcal{D}_c = \{D_{c_1}, D_{c_2}\} = \{\{a, b\}, \{c, d\}, \{e, f\}\}$$

$$C_c = \{c_{ace}, c_{acf}, p_{ade}, p_{adf}, p_{bce}, p_{bcf}, p_{bde}, p_{bdf}\}$$

L'ensemble des mots-clefs communs entre ces deux groupes est $\{a, c, d\}$. Les coefficients de G_c sont stockés dans un tableau associatif, la Figure 17 fournit une représentation graphique de cette structure de données.

Lors du parcours des processus amonts, nous déterminons la clef de hachage pour chaque processus, puis nous récupérons les coefficients qui peuvent leur être associés.

La clef de hachage d'un élément est calculée en ne conservant que ses mots-clefs qui sont présents dans la liste des mots-clefs communs. Comme les dimensions d'un groupe sont compatibles et que chaque élément d'un groupe est indexé par des mots-clefs de dimensions différentes, la clef de hachage d'un élément est donc forcément composée de mots-clefs communs entre les deux groupes. Mais certains éléments peuvent avoir une indexation commune avec les éléments de l'autre groupe sans pour autant être éligibles pour faire partie d'une relation détaillée.

Considérons, par exemple, un groupe de processus G_{p1} et un groupe de coefficients G_c tels que :

$$\begin{aligned} G_{p1} &= (\mathcal{D}_{p1}, P_{p1}) \\ &= (\{ \{a, b\}, \{c, d\} \}, P_{p1}) \end{aligned}$$

$$\begin{aligned} G_c &= (\mathcal{D}_c, C_c) \\ &= (\{ \{a, b\}, \{c, e\} \}, C_c) \end{aligned}$$

L'ensemble des mots-clefs communs entre les deux groupes est $\{a, b, c\}$. La clef de hachage de c_{ae} est donc a , or ce coefficient ne doit pas faire partie d'une relation détaillée car le nombre d'appariements entre ces deux groupes est égal

à 2 et le cardinal de l'intersection des mots-clefs de n'importe lequel des processus de G_{p1} avec les mots-clefs de c_{ae} est toujours égal à 1.

Pour calculer la clef de hachage d'un élément, il est donc nécessaire de déterminer si le nombre d'éléments qui appartiennent à cette clef est égal au nombre d'appariements.

Cette optimisation peut aussi être appliquée à la recherche de processus aval qui doivent être associés à un couple (processus amont, coefficient).

On détermine d'abord l'ensemble des mots-clefs communs entre le groupe de processus résultant de la multiplication de G_{p1} par G_c . Puis on indexe les processus aval en fonction de ces mots-clefs communs et du nombre d'appariements $\alpha(\mathcal{D}_{p1} \cup_D \mathcal{D}_c, \mathcal{D}_{p2})$.

Ici aussi, nous réduisons l'espace de recherche aux seuls processus aval éligibles pour une relation détaillée étant donné un processus amont et un coefficient. Et il n'est plus nécessaire de recourir à la seconde partie de la condition située ligne 5 de l'algorithme 3.

Il existe trois cas particuliers que nous devons traiter spécifiquement :

- il n'y a aucun mot-clef commun entre le groupe de processus amont et le groupe de coefficients. Dans ce cas nous avons $\alpha(\mathcal{D}_{G_{p1}}, \mathcal{D}_{G_c}) = 0$;
- il n'y a aucun mot-clef commun entre le résultat de la multiplication G_{p1} par G_c et le groupe de processus aval. Dans ce cas nous avons $\alpha(\mathcal{D}_{G_{p1}} \cup_D \mathcal{D}_{G_c}, \mathcal{D}_{G_{p2}}) = 0$;
- la combinaison des deux cas précédents. Dans ce cas nous avons $\alpha(\mathcal{D}_{G_{p1}}, \mathcal{D}_{G_c}) = 0$ et $\alpha(\mathcal{D}_{G_{p1}} \cup_D \mathcal{D}_{G_c}, \mathcal{D}_{G_{p2}}) = 0$.

Dans ces trois cas, la version naïve de l'algorithme crée des relations détaillées pour chaque combinaison d'éléments. Donc, dans la version optimisée, si le calcul de l'un des nombres d'appariements est égal à zéro, nous devons parcourir tous les éléments : soit du groupe de coefficients, soit du groupe de processus aval, soit des deux. C'est pourquoi nous rajoutons une clef de hachage permettant de regrouper tous les éléments d'un groupe si le nombre d'appariements est égal à zéro.

La version finale de l'algorithme de conversion est présentée dans le listing 4.

Si les groupes n'ont aucun mots-clef en commun il faut créer $|P_{G_{p1}}| * |P_{G_c}| * |P_{G_{p2}}|$ relations détaillées. En considérant que tous les groupes ont le même nombre d'éléments, nous avons une complexité dans le pire des cas identique à la version naïve en $O(n^3)$. Mais il est probable qu'une base de données basée sur notre approche ne contiendrait pas de macro-relations de ce type. La complexité moyenne doit donc être bien inférieure à $O(n^3)$. Dans la section suivante, nous présentons les résultats d'une expérimentation sur le passage à l'échelle de cet algorithme sur des jeux de données synthétiques. Les résul-

tats nous montrent que cette version optimisée nous permet d'améliorer les performances d'au moins deux ordres de grandeur.

Algorithme 4: Version optimisée de la conversion d'une macro-relation en un ensemble de relations détaillées

Entrées : G_{P1}, G_{P2}, G_c

```

1 relationsDétailées  $\leftarrow \emptyset$ 
2 dictCoeffs  $\leftarrow$  Dictionnaire des coefficients
3 dictProcessusAval  $\leftarrow$  Dictionnaire des processus avals
4 pour chaque  $p_1 \in G_{P1}$ .processus faire
5   | clefHachageCoeff  $\leftarrow$  Clef de hachage de  $p_1$ 
6   | si clefHachageCoeff not null alors
7   |   | pour chaque  $c \in \text{dictCoeffs}[clefHachageCoeff]$  faire
8   |   |   | clefHachageProcessusAval  $\leftarrow$  Clef de hachage de  $p_1$ 
9   |   |   | si clefHachageProcessusAval not null alors
10  |   |   |   | pour chaque  $p_2 \in \text{dictProcessusAval}[clefHachageProcessusAval]$ 
11  |   |   |   |   | faire
12  |   |   |   |   | relationsDétailées +=  $((p_1, p_2), c)$ 
12 retourner relationsDétailées
```

3.5 EXPÉRIMENTATION SUR LE PASSAGE À L'ÉCHELLE DE L'ALGORITHME DE CONVERSION

Nous avons réalisé des tests de passage à l'échelle de cet algorithme pour :

- vérifier l'impact des optimisations apportées à la version naïve ;
- nous assurer que nous pouvons convertir différents types de macro-relations de taille variables dans un temps raisonnable⁹.

Pour cette expérimentation, nous avons utilisé un jeu de données synthétiques. Nous avons généré des groupes de processus et de coefficients de taille croissante. Les résultats de cette expérimentation sont visibles dans les figures 18 à 22.

Au préalable, quelques remarques s'appliquent à toutes les expériences :

- nous n'avons étudié le passage à l'échelle que pour des groupes de processus contenant au maximum 10 000 processus. Cette limite est dans l'ordre de grandeur de la base de données d'inventaires la plus importante, en termes de volumétrie, qui contient un peu plus de 4 000 processus [28] ;
- cette expérimentation a été réalisée avec des groupes simplifiés à leur plus simple expression : il n'y a pas de processus ou de coefficients mais juste des mots-clés, l'objectif principal étant de tester le passage à l'échelle sur le traitement des coordonnées des éléments à associer dans des relations détaillées ;

9. L'approche proposée ayant vocation à être intégrée dans une application SaaS, l'algorithme de conversion doit être le plus efficace possible.

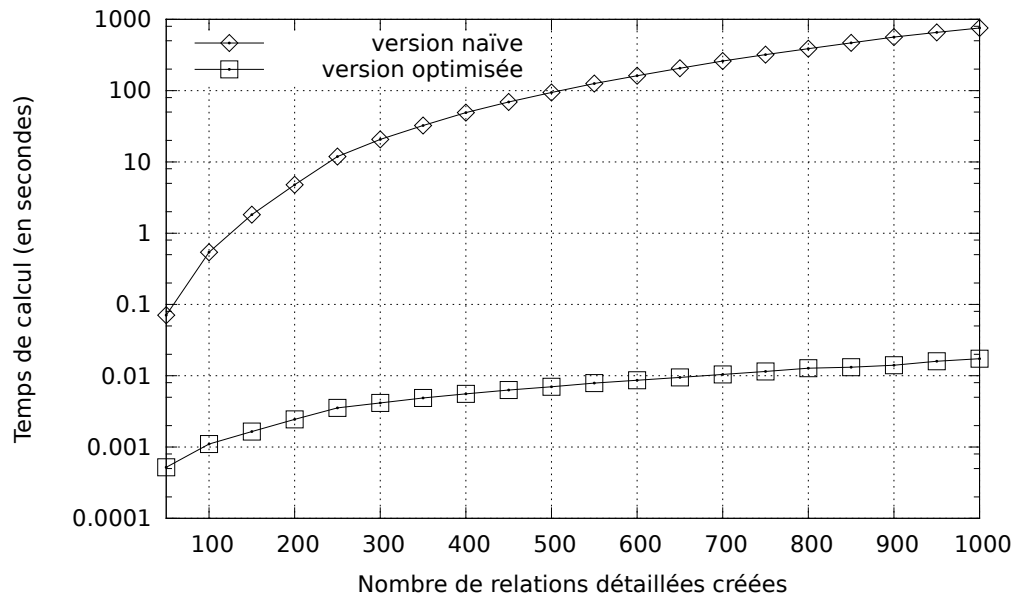


FIGURE 18: Comparaison du temps de calcul pour convertir des macro-relations de taille croissantes entre la version naïve et la version optimisée de l'algorithme de conversion des macro-relations. L'échelle de l'axe des ordonnées est logarithmique.

- presque tous les graphiques utilisent une échelle logarithmique pour l'axe des ordonnées représentant le temps de calcul ; ceci afin de visualiser plus facilement les différences d'ordre de grandeur ;
- l'expérimentation a été réalisée en PHP¹⁰ sous Linux sur un PC équipé d'un Intel Xéon E31245 à 3.30GHz avec 8Go de mémoire vive.

La première expérience a été réalisée pour déterminer l'impact des optimisations apportées à la version naïve de l'algorithme de conversion. Nous avons généré un groupe de processus amont avec un nombre croissant de mots-clefs contenus dans une seule dimension. Les coordonnées des cellules de ce groupe ne comportant qu'un seul mot-clef, nous avons autant de processus que de mot-clefs dans la dimension. Nous ne créons que des relations normales : les groupes de coefficients et de processus aval sont des copies du groupe de processus amont. Il y a donc autant de relations détaillées créées que de processus dans le groupe amont.

Dans le graphique présenté figure 18, nous avons tracé le temps de calcul nécessaire aux deux versions de l'algorithme en fonction du nombre de processus dans le groupe amont. Nous avons limité le nombre de processus amonts à 1 000 car le temps de calcul des macro-relations pour la version naïve devenait trop important. Malgré cette limitation, nous pouvons constater qu'il y a déjà cinq ordres de grandeur de moins pour la version optimisée de l'algorithme par rapport à la version naïve pour traiter des groupes contenant 1000 éléments. Pour des groupes de faibles tailles (moins de 100 éléments), nous avons au moins deux ordres de grandeur d'écart entre les deux versions de

10. Pour être conforme aux choix technologiques de My C-sense.

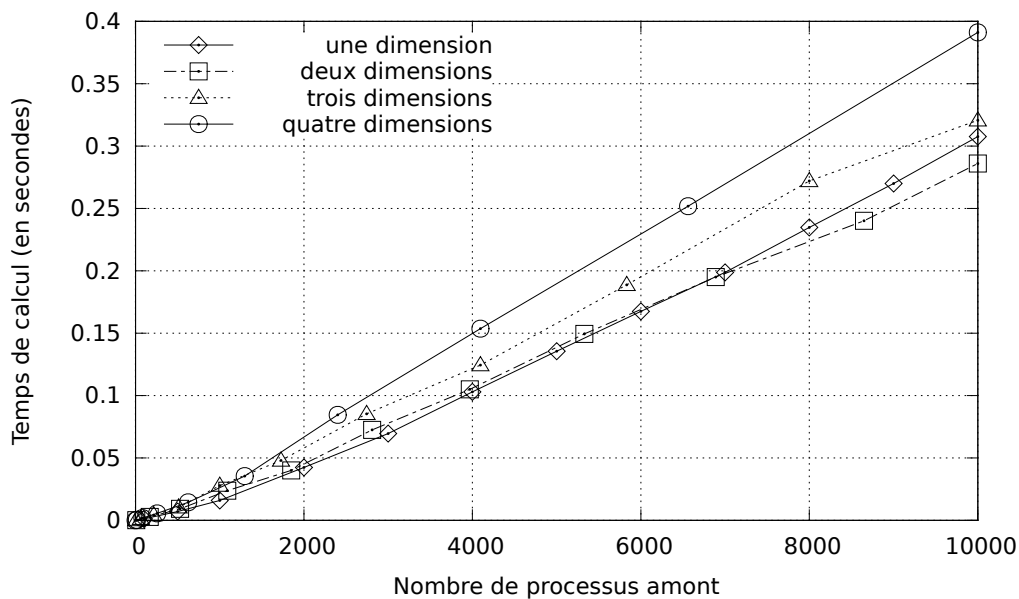


FIGURE 19: Comparaison du temps de calcul pour convertir des macro-relations de taille croissante avec la version optimisée de l'algorithme de conversion pour des relations impliquant des groupes de une, deux, trois et quatre dimensions.

l'algorithme de conversion.

La deuxième expérience porte sur le temps de calcul pour convertir des macro-relations impliquant des groupes de plus d'une dimension. Nous avons généré des groupes avec une, deux, trois et quatre dimensions. Pour chacun des cas, nous générons le groupe amont puis créons des copies des dimensions de ce groupe pour le groupe de coefficients et le groupe de processus aval. Nous créons donc des macro-relations normales uniquement. Le résultat de cette expérience est présenté dans la figure 19. Nous constatons qu'il n'y a pas de différence notable entre le traitement de macro-relations quel que soit le nombre de dimensions des groupes impliqués.

La troisième expérience est centrée sur la conversion de macro-relations projetées. Nous convertissons des macro-relations impliquant un groupe de processus amont avec un seul processus, un groupe de processus aval contenant un nombre croissant de processus. Nous avons testé deux variantes : une avec un groupe de coefficients contenant un seul coefficient et une autre avec un groupe de coefficients contenant un nombre croissant de coefficients (des groupes de coefficients avec les mêmes dimensions que le groupe de processus aval). Les résultats de cette expérience sont présentés dans la figure 20. Nous constatons qu'il y a une différence d'environ un ordre de grandeur entre les deux variantes. Cette différence est simplement due aux pré-traitements que l'on doit réaliser sur le groupe de coefficients : création de la table de hachage, calcul de l'union de dimensions et calcul du nombre d'appariements.

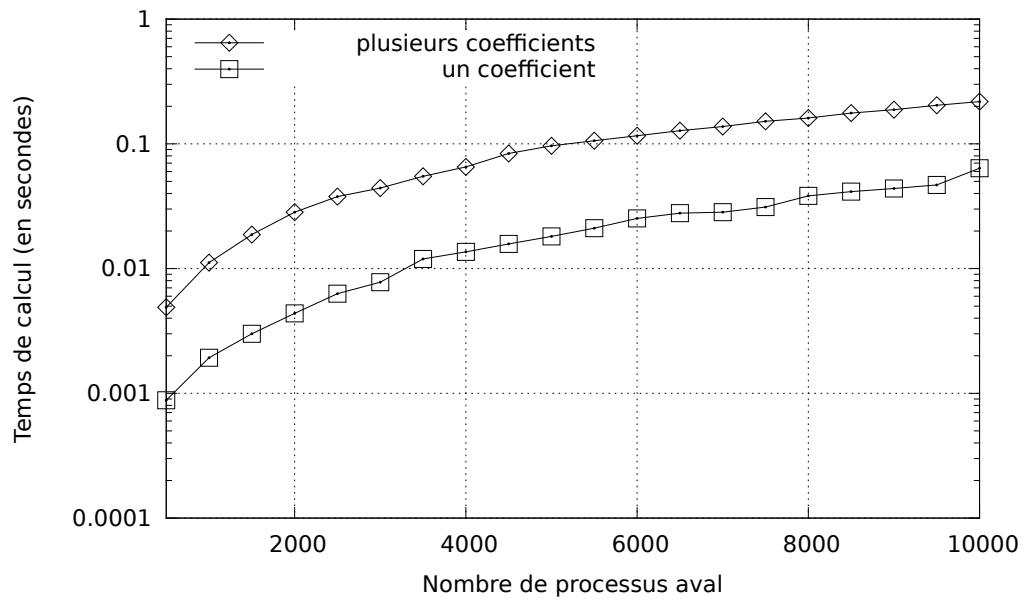


FIGURE 20: Comparaison du temps de calcul pour convertir des macro-relations projetées de taille croissante avec la version optimisée de l'algorithme de conversion pour des relations impliquant un groupe de coefficients ou un seul coefficient. L'échelle de l'axe des ordonnées est logarithmique.

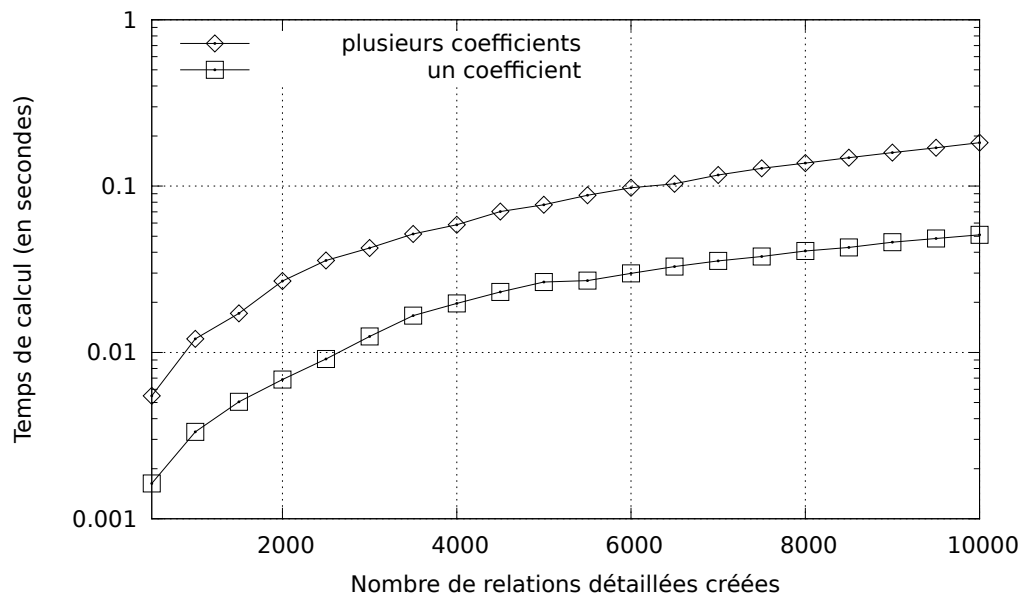


FIGURE 21: Comparaison du temps de calcul pour convertir des macro-relations agrégées de taille croissante avec la version optimisée de l'algorithme de conversion pour des relations impliquant un groupe de coefficients ou un seul coefficient. L'échelle de l'axe des ordonnées est logarithmique.

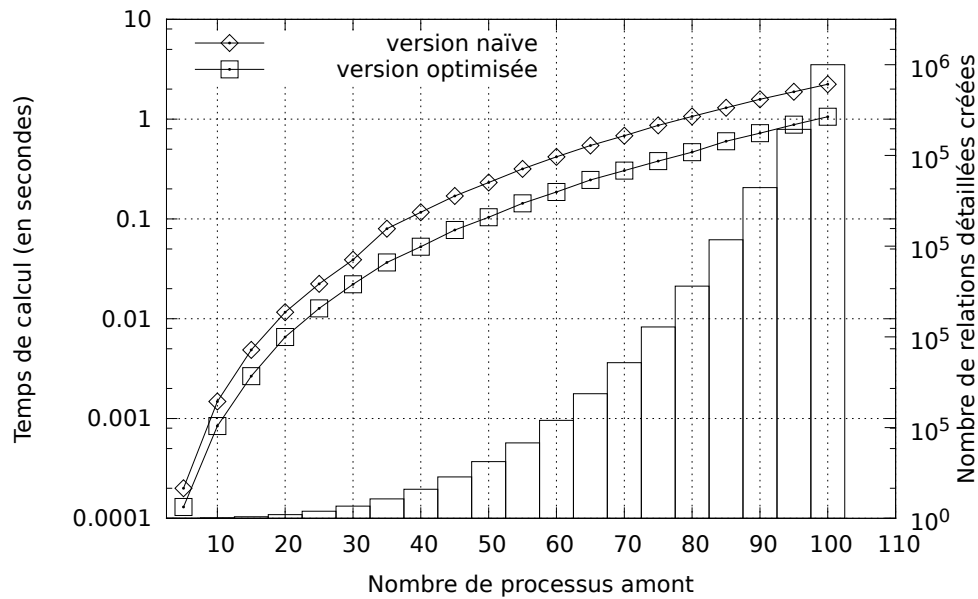


FIGURE 22: Comparaison du temps de calcul pour convertir des macro-relations de taille croissante impliquant des groupes n'ayant aucun mot-clef en commun entre la version optimisée et la version naïve de l'algorithme de conversion. Ce graphique présente aussi le nombre de relations détaillées créées en fonction de la taille des groupes composant la macro-relation. L'échelle de l'axe des ordonnées pour le temps de calcul est logarithmique.

La quatrième expérience est l'équivalent de la troisième sur la conversion de macro-relations agrégées. Nous convertissons des macro-relations impliquant un groupe de processus amont avec un nombre croissant de processus, un groupe de processus aval contenant un seul processus. Comme pour l'expérience sur les macro-relations projetées, nous avons testé deux variantes : une avec un groupe de coefficients ne contenant qu'un seul coefficient et une autre avec un groupe de coefficients contenant un nombre croissant de coefficients. Les résultats de cette expérience sont présentés dans la figure 21. Ici aussi, nous constatons qu'il y a une faible différence entre les deux variantes. Elle s'explique aussi par les pré-traitements à réaliser sur le groupe de coefficients.

La dernière expérience porte sur la conversion de macro-relations n'ayant aucun mot-clef en commun, *i.e.*: qui n'ont aucune dimension ayant une intersection non nulle. Avec cette expérience, nous testons le pire des cas pour l'algorithme optimisé car il faut créer des relations détaillées correspondant à chaque combinaison de processus amont, coefficient et processus aval. Il est donc nécessaire de parcourir toutes les combinaisons d'éléments, ce que fait la version naïve de l'algorithme. Les résultats de cette expérience sont présentés dans la figure 22.

Pour de telles macro-relations, nous avons $|P_{G_{p_1}}| * |P_{G_c}| * |P_{G_{p_2}}|$ relations détaillées à créer. Nous avons limité cette expérience à des groupes de processus amont (ainsi que les groupes de coefficients et de processus aval qui sont des copies du groupe amont) ne contenant pas plus de 100 mots-clefs. Au delà de

cette limite, le nombre de relations détaillées à créer devient beaucoup trop conséquent. L'expérience se retrouve limitée par l'occupation mémoire de ces relations détaillées créées. La figure 22 présente aussi le nombre de relations détaillées créées en fonction de la taille du groupe amont.

Nous constatons que la version optimisée de l'algorithme rejoint alors le temps de traitement requis par la version naïve. Dans un cas réel, nous n'obtiendrons vraisemblablement pas ce cas de figure car il signifierait que le macro-graphe contiendrait des macro-relations entre des groupes de processus sans lien sémantique. De plus, les matrices de flux entre les processus dans les bases de données d'inventaire existantes sont relativement creuses. Nous aurions donc probablement beaucoup de coefficients nuls dans des groupes de coefficients impliqués dans de telles macro-relations. Il est alors possible de ne pas créer de relations détaillées utilisant des coefficients nuls, donc de réduire l'occupation mémoire nécessaire.

3.6 SYNTHÈSE

Nous avons proposé une méthode de modélisation sémantique d'une base de données d'inventaires en cycle de vie qui permet :

- de proposer une vue d'ensemble des relations de dépendance entre les processus ;
- de simplifier la gestion de ces relations de dépendance.

Dans ce chapitre nous avons présenté le formalisme de notre méthode, les algorithmes nécessaires pour la conversion des macro-relations en relations détaillées et une étude sur leur passage à l'échelle.

Nous avons finalement appliqué deux principes de programmation aux bases de données d'inventaire en cycle de vie :

- Keep It Simple and Stupid : nous présentons à l'utilisateur final une modélisation plus simple à comprendre ;
- Don't Repeat Yourself : nous réduisons le nombre de relations de dépendance que doit créer (et maintenir) un éditeur de la base.

Mais, pour l'éditeur d'une base d'inventaires reprenant cette approche, l'utilisation d'une ontologie complexifie son travail ¹¹.

Dans le chapitre suivant nous présentons une étude de cas de l'application de cette méthode sur la base de données d'inventaires du NREL [64].

11. Contrairement à l'utilisateur final, il ne bénéficie pas du principe K.I.S.S

ÉTUDE DE CAS DE LA MODÉLISATION MULTI-NIVEAUX

Dans ce chapitre nous proposons une application de l'approche multi-niveaux sur une base de données d'inventaires en cycle de vie existante. Nous avons choisi d'utiliser la base de données maintenue par le National Renewable Energy Laboratory (NREL) [64]. Cette base contient des processus pour un ensemble d'activités économiques aux États-Unis. Elle contient environ 600 processus accessibles gratuitement depuis une interface web. Les données d'inventaires pour chaque processus peuvent être exportées dans des fichiers Excel ou XML respectant le format Ecospold [41]. En plus d'inclure des métadonnées pour chaque processus (sources, commentaires, etc.), ces deux formats incluent les relations de dépendance entre les processus ainsi que les coefficients de dépendance.

Dans la première section nous présentons les limites de notre étude. Dans la deuxième section nous détaillons trois macro-relations puis nous discutons sur les alternatives à la modélisation présentée.

4.1 LIMITES DE L'ÉTUDE : PRODUCTION D'ÉLECTRICITÉ AUX ETATS-UNIS

Nous avons choisi de restreindre notre étude à la production d'électricité. Les données environnementales de la production d'électricité sont découpées pour les 27 sous-régions énergétiques du territoire américain [64]. Ces sous-régions sont définies par l'Environmental Protection Agency et la Generation Resource Integrated Database (eGRID). Chaque sous-région de l'eGRID représente une portion du territoire américain qui a des mix énergétiques similaires et qui est potentiellement isolée en raison de contraintes de transmission.

Certains processus de cette base de données ne sont pas détaillés (et marqués comme "factices"); nous n'avons donc pas un jeu de données complet en termes de relations de dépendance. Mais même sans ces processus, il reste suffisamment complexe pour illustrer notre proposition.

La figure 23 présente le graphe détaillé correspondant à ce jeu de données limitées à 7 sous-régions de l'eGRID et à une profondeur de deux niveaux maximum. Ce graphe détaillé contient 27 processus et 72 relations de dépendance.

4.2 MACRO-GRAPHE ET MACRO-RELATIONS

Nous proposons de créer un macro-graphe qui mette en avant les grandes familles de ressources utilisées pour produire de l'électricité. Nous pouvons modéliser ces données dans un macro-graphe présenté dans la figure 24. Ce

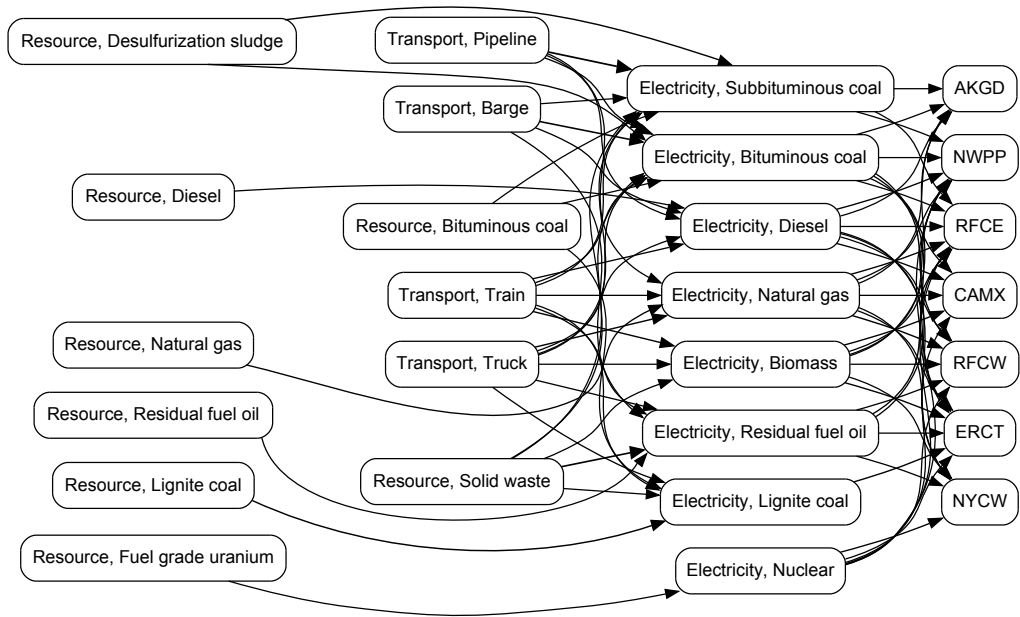


FIGURE 23: Graphe détaillé correspondant au jeu de données extrait de la base de données d’inventaires en cycle de vie du NREL. Les groupes de coefficients ne sont pas représentés pour simplifier la figure.

graphe contient 13 groupes de processus et 17 macro-relations. Nous avons donc besoin, a priori, de 17 groupes de coefficients. Dans les sections suivantes, nous détaillons quelques regroupements et macro-relations que nous utilisons pour obtenir ce macro-graphe.

4.2.1 Production d’électricité à partir du charbon

Nous créons une macro-relation entre la production d’électricité à partir de charbon et la production d’électricité dans les sous-régions de l’eGRID. Le groupe de processus G_{p1} contient les processus de production d’électricité à partir de charbon. Ce groupe est composé d’une dimension contenant les différents types de charbon utilisés pour produire de l’électricité. Le groupe de processus G_{p2} contient les processus de production d’électricité dans les sous-régions de l’eGrid. Ce groupe est composé d’une dimension contenant les sous-régions de l’eGrid.

La macro-relation impliquant ces deux groupes requiert un groupe de coefficients G_c contenant des coefficients qui indiquent quelle est la quantité d’électricité produite à partir de charbon dans chacune des sous-régions de l’eGrid. Ce groupe est composé de deux dimensions : une contenant les sous-régions de l’eGrid et une contenant les différents types de charbon utilisés pour produire de l’électricité. Les groupes impliqués dans cette macro-relation et les relations détaillées obtenues après sa conversion sont présentés dans la figure 25. Certaines sous-régions n’ont pas recours à certains types de charbon pour produire de l’électricité. Nous avons donc certains coefficients qui ont une va-

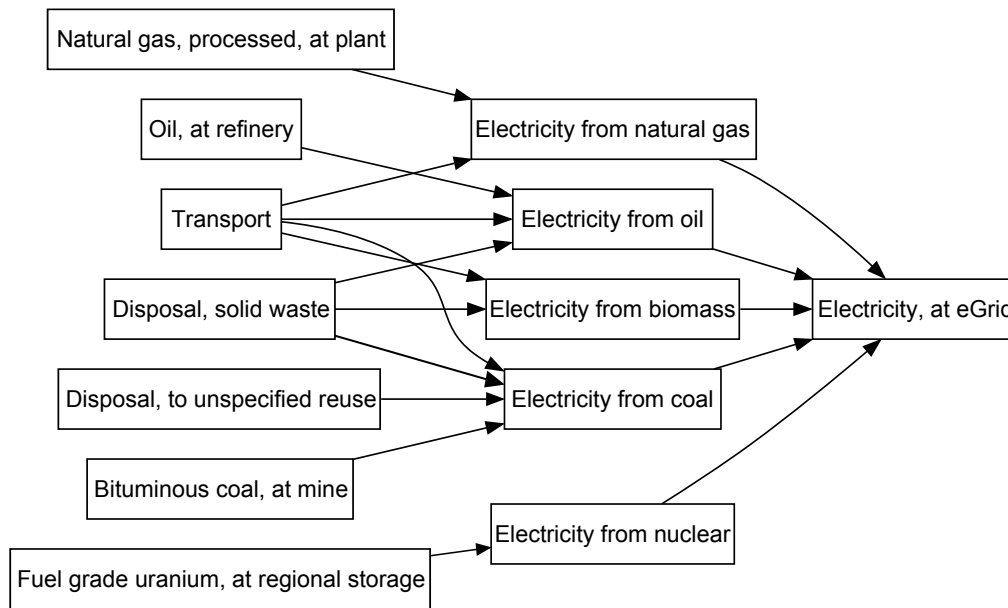


FIGURE 24: Macro-graphe correspondant au jeu de données extrait de la base de données d’inventaires en cycle de vie du NREL. Les coefficients ne sont pas représentés pour simplifier la figure.

leur nulle, ce qui induit que toutes les relations détaillées ne seront pas créées. Ceci est le cas pour la sous-région *AKGD* qui n’utilise pas de centrale à lignite.

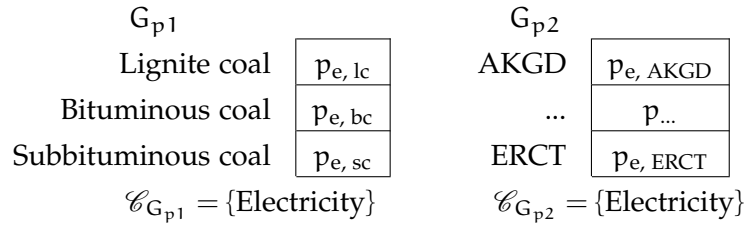
4.2.2 Moyens de transport utilisés pour la production d’électricité à partir de biomasse

Nous créons une macro-relation entre les processus de transport de marchandises et le processus de production d’électricité à partir de biomasse. La figure 26 présente les groupes impliqués dans cette macro-relation et les relations détaillées obtenues après l’avoir convertie. Le groupe de processus G_{p1} contient les processus de transport de marchandises. Il est composé d’une dimension contenant les différents types de transports. Le groupe de processus G_{p2} contient le processus de production d’électricité à partir de biomasse.

La macro-relation impliquant ces deux groupes requiert un groupe de coefficients G_c contenant des coefficients qui indiquent l’utilisation des différents modes de transport pour produire de l’électricité à partir de biomasse. La dimension contenant les modes de transport de G_c comporte uniquement les mots-clefs *Truck* et *Train*, des trains et des camions étant seulement utilisés. Au final, seuls certains processus du groupe amont seront utilisés, ce groupe pouvant être intégralement utilisé dans d’autres macro-relations.

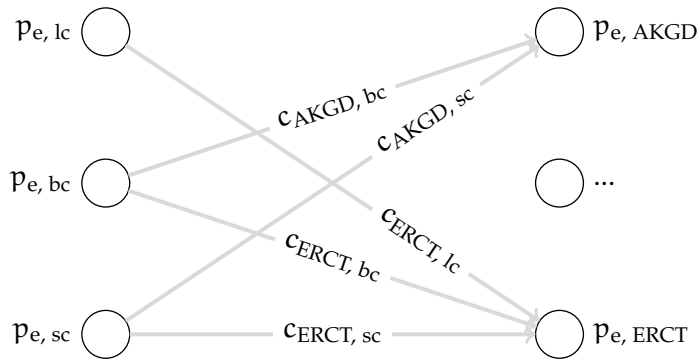
4.2.3 Ressources utilisées pour la production d’électricité à partir de pétrole

Nous créons une macro-relation entre les processus de production de produits pétroliers et les processus de production d’électricité à partir du pétrole. La figure 27 présente les groupes impliqués dans cette macro-relation et les re-



G_c	AKGD	...	ERCT
Lignite coal	$c_{AKGD,lc}$...	$c_{ERCT,lc}$
Bituminous coal	$c_{AKGD,bc}$...	$c_{ERCT,bc}$
Subbituminous coal	$c_{AKGD,sc}$...	$c_{ERCT,sc}$

(a) Groupes de la macro-relation



(b) Relations détaillées

FIGURE 25: Conversion en relations détaillées d’une macro-relation $((G_{p1}, G_{p2}), G_c)$. G_{p1} contient les processus de production d’électricité à partir de charbon. G_{p2} contient les processus de production d’électricité pour les sous-régions de l’eGrid. G_c contient les coefficients de dépendance des sous-régions de l’eGrid vis à vis de la production d’électricité à partir du charbon. La notation de l’indexation des éléments est simplifiée : $p_{e,lc}$ correspond au processus $p^{\text{Electricity, Lignite coal}}$.

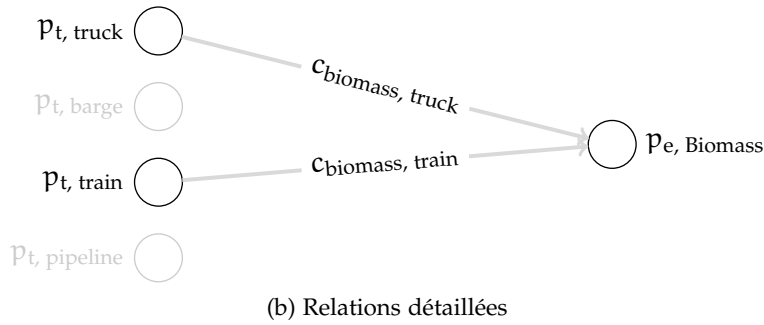
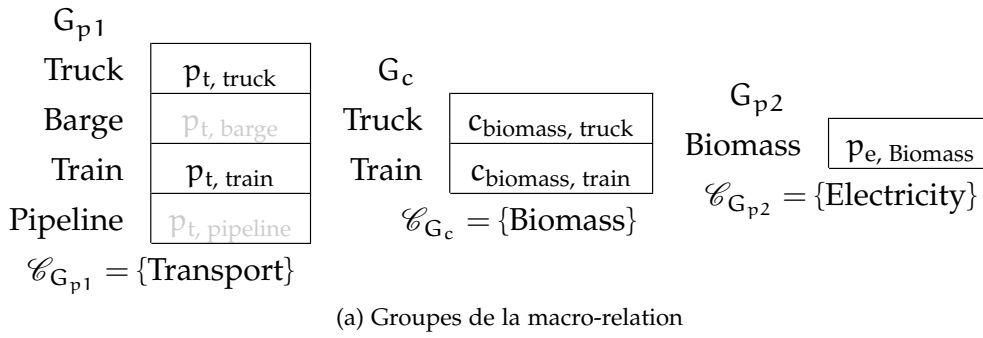


FIGURE 26: Conversion en relations détaillées d’une macro-relation entre les processus de transport de marchandises et le processus de production d’électricité à partir de biomasse.

lations détaillées obtenues après l’avoir convertie. Le groupe G_{p1} contient les processus de production de produits pétroliers, le groupe G_{p2} contient le processus de production d’électricité à partir de pétrole et le groupe G_c contient des coefficients qui indiquent quelle est la quantité de produits pétroliers utilisés pour produire de l’électricité à partir de pétrole. Ces trois groupes sont composés d’une même dimension contenant les différents produits pétroliers utilisés pour produire de l’électricité et ne diffèrent que par leurs mots-clefs communs.

4.3 MACRO-MODÉLISATIONS ALTERNATIVES

Il est aussi possible de réduire le nombre de groupes de coefficients nécessaires à 12. Par exemple, le groupe de processus ds modes de transport peut être relié à chaque groupe de processus de production d’électricité à l’aide d’un groupe de coefficients regroupant tous les coefficients de dépendance entre les différents modes de transport et les processus de production d’électricité (d’une façon similaire à l’illustration de la méthode de conversion telle que présentée dans la Figure 9). Nous pouvons aussi utiliser seulement 8 groupes de coefficients si nous stockons dans un seul groupe tous les coefficients de dépendances entre les différents processus de production d’électricité et le groupe contenant les processus de production d’électricité dans les différentes régions de l’eGRID.

Il existe plusieurs macro-graphes correspondant à ce jeu de données. Nous pouvons, par exemple, créer un macro-graphe dans lequel nous regroupons les

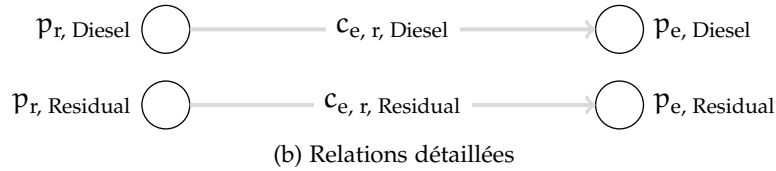
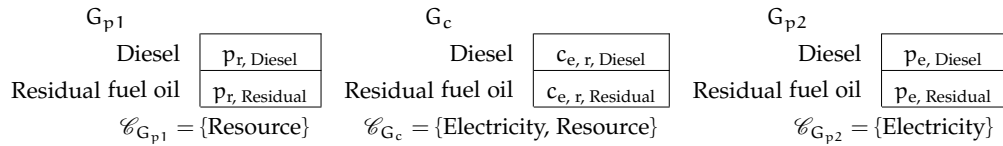


FIGURE 27: Conversion en relations détaillées d'une macro-relation entre les processus d'extraction de pétrole et les processus de production d'électricité à partir de pétrole.

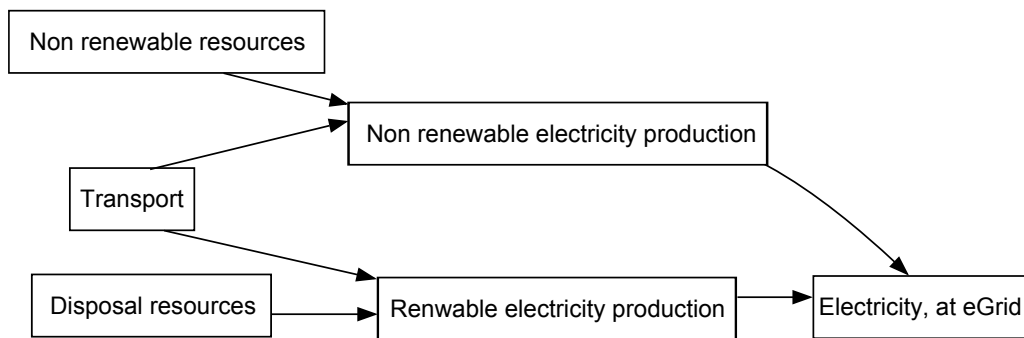


FIGURE 28: Alternative de macro-graphe correspondant au jeu de données extrait de la base de données d'inventaires en cycle de vie du NREL. Les coefficients ne sont pas représentés pour simplifier la figure.

moyens de production d'électricité renouvelables dans un groupe et les non renouvelables dans un autre. La Figure 28 présente le macro-graphe correspondant à cette modélisation alternative. Il est aussi possible de regrouper les sous-régions de l'eGrid en fonction de leur emplacement géographique.

4.4 SYNTHÈSE

L'étude de cas que nous avons réalisée sur une base d'inventaires existante nous a permis de valider la pertinence du regroupement sémantique des processus pour en faciliter la compréhension.

APPROCHES DÉCLARATIVES DE LA MODÉLISATION MULTI-NIVEAUX D'UN INVENTAIRE

Dans cette partie nous étudions deux approches déclaratives de notre modélisation des bases de données d'inventaires : une approche basée sur l'algèbre relationnelle pour la modélisation multi-niveaux et une approche basée sur les outils du web sémantique pour répondre à des requêtes sémantiques sur la composition des processus.

Ces approches sont utiles pour proposer aux utilisateurs de notre méthode des possibilités d'implémentation utilisant des théories et des outils bien connus. L'approche basée sur l'algèbre relationnelle peut être implémentée sur un Système de Gestion de Base de Données Relationnelle (SGBDR) utilisant SQL [55] et permet de mettre à profit les fonctionnalités de ces systèmes. L'utilisation des outils du web sémantique permet d'inférer des connaissances sur un modèle OWL [54] de la base et d'interroger le graphe détaillé avec SPARQL [73].

Dans la première section nous présentons l'approche basée sur l'algèbre relationnelle puis son implémentation en SQL dans la seconde section. Dans la troisième section nous présentons une modélisation en OWL du graphe détaillé que nous exploitons à l'aide du langage de requêtes SPARQL pour répondre à des requêtes sémantiques dans la dernière section.

5.1 MODÉLISATION MULTI-NIVEAUX BASÉE SUR L'ALGÈBRE RELATIONNELLE

Dans cette section nous présentons l'utilisation de l'algèbre relationnelle à la modélisation multi-niveaux d'un inventaire en cycles de vie et notamment une méthode basée sur l'algèbre relationnelle pour convertir des macro-relations en relations détaillées.

Nous avons besoin de deux primitives pour réaliser la conversion d'une macro-relation en relations détaillées :

- la multiplication d'un groupe de processus par un groupe de coefficients ;
- l'addition de deux groupes de processus.

Pour simplifier la présentation de cette approche nous considérons qu'un processus n'a qu'un flux scalaire vers l'environnement et que nous calculons directement les flux cumulés¹. Nous modélisons un groupe de processus ou de coefficients par une relation possédant un attribut par dimension et un attri-

1. Cette simplification a comme objectif de faciliter la lecture. L'implémentation SQL que nous présentons dans la section suivante ne reprend pas cette simplification.

but pour la valeur scalaire du flux du processus ou pour la valeur du coefficient.

Exemple : soit G_{p1} un groupe contenant les processus de transport de marchandises, G_{p2} un groupe contenant des processus de production d'électricité et G_c un groupe contenant des coefficients de dépendance entre ces deux groupes. Ils sont définis tels que :

$$\begin{aligned} G_{p1} &= (\{ D_{p1_1} \}, P_{p1}) = (\{ \{ \text{Truck, Barge} \} \}, P_{p1}) \\ G_{p2} &= (\{ D_{p1_2} \}, P_{p2}) = (\{ \{ \text{Oil, Coal} \} \}, P_{p2}) \\ G_c &= (\{ D_{c_1}, D_{c_2} \}, C_c) = (\{ \{ \text{Truck, Barge} \}, \{ \text{Oil, Coal} \} \}, C_c) \end{aligned}$$

Nous les représentons en utilisant les trois relations suivantes :

G_{p1}	D_{p1_1}	value		G_{p2}	D_{p1_2}	value
	Truck	V_{Truck}			Oil	V_{Oil}
	Barge	V_{Barge}			Coal	V_{Coal}

G_c	D_{c_1}	D_{c_2}	value
	Truck	Oil	$V_{\text{Truck, Oil}}^c$
	Truck	Coal	$V_{\text{Truck, Coal}}^c$
	Barge	Oil	$V_{\text{Barge, Oil}}^c$
	Barge	Coal	$V_{\text{Barge, Coal}}^c$

Le calcul des impacts d'une macro-relation se décompose en deux étapes :

1. les processus du groupe amont sont multipliés par les coefficients du groupe de coefficients. Nous obtenons un groupe de processus temporaires (que nous nommons $G_{\text{temporaire}}$).
2. les impacts des processus du groupe $G_{\text{temporaire}}$ sont sommés au groupe de processus aval.

5.1.1 Multiplication d'un groupe de processus par un groupe de coefficients

La première étape se fait en réalisant une jointure conditionnelle entre G_{p1} et G_c sur les dimensions communes (c'est-à-dire les dimensions ayant une intersection non nulle²) et une multiplication de l'impact des processus par les coefficients. Nous introduisons un opérateur noté Π -jointure entre deux groupes. Nous notons une jointure conditionnelle $\bowtie_{[C]}$ où C est la condition qui exprime quels attributs des relations impliquées dans la jointure doivent être égaux.

Définition 10. Soit G_1 et G_2 deux groupes. On note $G_1 \bowtie_{\Pi} G_2$ la Π -jointure entre G_1 et G_2 telle que :

$$G_1 \bowtie_{\Pi} G_2 = \Pi_{[D_{p1}, \dots, D_{pn}, V^* V']}(G_1 \bowtie_{[C]} G_2)$$

2. Comme défini dans le chapitre [MODÉLISATION MULTI-NIVEAUX D'UN INVENTAIRE EN CYCLE DE VIE](#).

Cette opération nécessite une condition pour la jointure que nous nommons *condition d'appariement*. Pour la déterminer nous avons besoin de savoir quelles sont les dimensions des deux groupes qui ont des éléments en commun.

Nous définissons la notion d'ensemble d'appariements³ :

Définition 11. Soit G_1 et G_2 deux groupes et \mathcal{D}_1 et \mathcal{D}_2 leurs ensembles de dimensions respectifs. On note $A(G_1, G_2)$ l'ensemble d'appariements entre ces deux groupes qui contient des paires ordonnées de dimensions telles que :

$$A(G_1, G_2) = \{(D_1, D_2) \in \mathcal{D}_1 \times \mathcal{D}_2 \mid D_1 \cap D_2 \neq \emptyset\}$$

Exemple : soit G_{p1} et G_{p2} deux groupes de processus tels que :

$$\begin{aligned} G_{p1} &= (\mathcal{D}_{G_{p1}}, P_{G_{p1}}) \\ &= (\{D_{G_{p11}}, D_{G_{p12}}\}, P_{G_{p1}}) \\ &= (\{\{a, b\}, \{c, d\}\}, P_{G_{p1}}) \\ G_{p2} &= (\mathcal{D}_{G_{p2}}, P_{G_{p2}}) \\ &= (\{D_{G_{p21}}, D_{G_{p22}}\}, P_{G_{p2}}) \\ &= (\{\{c, d, e\}, \{b, h\}\}, P_{G_{p2}}) \end{aligned}$$

$$\text{Alors } A(G_{p1}, G_{p2}) = \{(D_{G_{p11}}, D_{G_{p22}}), (D_{G_{p12}}, D_{G_{p21}})\}.$$

La condition de la jointure est créée en utilisant l'ensemble d'appariements A telle que le premier élément de chaque paire de A doit être égal au second élément. Par exemple, si on a $A = \{(D_1, D_2), (D_3, D_4)\}$, alors la condition est égale à " $D_1 = D_2$ & $D_3 = D_4$ ".

On note P le résultat de cette opération tel que $P = (\mathcal{D}_P, P_P) = (\{D_{p1}, \dots, D_{pn}\}, P_P)$. L'ensemble \mathcal{D}_P est composé de toutes les dimensions des deux groupes.

En considérant, par exemple, la Π -jointure entre le groupe de processus amont G_{p1} et le groupe de coefficients G_c présenté au début de cette section. On a :

$$A(G_{p1}, G_c) = \{(D_{p11}, D_{c1})\}$$

Donc la condition de la jointure conditionnelle est " $D_{p11} = D_{c1}$ " et la Π -jointure entre ces deux groupes s'exprime :

$$G_{p1} \bowtie_{\Pi} G_c = \Pi_{[D_{p11}, D_{c1}, D_{c2}, G_{p1.value} * G_{p2.value}]}(G_{p1} \bowtie_{[A]} G_c)$$

Le résultat de l'opération $G_{p1} \bowtie_{\Pi} G_c$ est le groupe de processus P tel que :

$$\begin{aligned} P &= (\mathcal{D}_P, P_P) \\ &= (\{D_{p1}, D_{p2}, D_{p3}\}, P_P) \\ &= (\{\{Truck, Barge\}, \{Truck, Barge, \{Oil, Coal\}\}\}, P_P) \end{aligned}$$

3. Cette notion est comparable à la notion de nombre d'appariements présentée dans le chapitre précédent.

Les dimensions D_{P1} et D_{P2} sont identiques car nous reprenons toute les dimensions de deux groupes de la Π -jointure. Finalement, le contenu de cette relation est :

P	D_{P1}	D_{P2}	D_{P3}	value
	Truck	Truck	Oil	$V_{Truck} * V_{Truck, Oil}^c$
	Truck	Truck	Coal	$V_{Truck} * V_{Truck, Coal}^c$
	Barge	Barge	Oil	$V_{Barge} * V_{Barge, Oil}^c$
	Barge	Barge	Coal	$V_{Barge} * V_{Barge, Coal}^c$

5.1.2 Addition de deux groupes de processus

La deuxième étape nécessaire pour traiter une macro-relation est l'addition du groupe temporaire P au groupe de processus aval G_{p2} . Cette opération est presque identique à la Π -jointure en remplaçant la multiplication des valeurs de G_{p1} et G_c par une addition des valeurs de P et G_{p2} . L'ensemble d'appariements est déterminé entre P et G_{p2} . Nous introduisons un nouvel opérateur nommé Σ -jointure entre deux groupes :

Définition 12. Soit G_1 et G_2 deux groupes et soit C_A la condition d'appariement entre ces deux groupes. On note $G_1 \bowtie_{\Sigma} G_2$ la Σ -jointure entre G_1 et G_2 telle que :

$$G_1 \bowtie_{\Sigma} G_2 = \Pi_{[D_1, \dots, D_n, V_1 + V_2]}(G_1 \bowtie_{[C_A]} G_2)$$

Mais cette opération ne permet pas de traiter les macro-relations agrégées que nous avons identifiées dans le chapitre **MODÉLISATION MULTI-NIVEAUX D'UN INVENTAIRE EN CYCLE DE VIE**. Pour traiter ce type de macro-relation, nous avons besoin d'effectuer une agrégation de certains tuples du groupe de processus temporaire P .

Considérons, par exemple, l'addition entre un groupe de processus temporaire P et un groupe de processus G . Ces groupes sont définis comme suit :

$$\begin{aligned} P &= (\mathcal{D}_P, P_P) \\ &= (\{ D_{P1}, D_{P2} \}, P_P) \\ &= (\{ \{a, b\}, \{c, d\} \}, P_P) \\ G &= (\mathcal{D}_G, P_G) \\ &= (\{ D_{G1}, D_{G2} \}, P_G) \\ &= (\{ \{a, e\}, \{f, g\} \}, P_G) \end{aligned}$$

Nous les représentons en utilisant les deux relations suivantes :

P	D_{P1}	D_{P2}	value	G	D_{G1}	D_{G2}	value
	a	c	V_{ac}		a	f	V_{af}
	a	d	V_{ad}		a	g	V_{ag}
	b	c	V_{bc}		e	f	V_{ef}
	b	d	V_{bd}		e	g	V_{eg}

En appliquant une Σ -jointure entre ces deux groupes, nous obtenons une relation qui présente des doublons au niveau de l'indexation :

$P \bowtie_{\Sigma} G$	D_{P1}	D_{P2}	value
	a	f	$V_{ac} + V_{af}$
	a	g	$V_{ac} + V_{ag}$
	a	f	$V_{ad} + V_{af}$
	a	g	$V_{ad} + V_{ag}$

Si P a plus de dimensions que G alors il faut agréger les valeurs de P pour les dimensions qui n'ont pas de mots-clés communs avec les dimensions de G (D_{P2} dans notre exemple). Nous devons donc agréger les valeurs de P en amont de la Σ -jointure.

Pour cela, nous devons déterminer quelles sont les dimensions de P qui ont une intersection nulle avec les dimensions de G . Lors du calcul de l'ensemble d'appariements entre P et G , nous connaissons les dimensions de P qui ont une intersection non nulle avec une dimension de G . Cet ensemble se note \mathcal{D}_P^A . L'ensemble $\mathcal{D}_P - \mathcal{D}_P^A$ contient les dimensions que nous souhaitons. C'est le complémentaire de \mathcal{D}_P^A par rapport à \mathcal{D}_P , aussi nous le notons $\overline{\mathcal{D}_P^A}$.

Dans notre exemple, nous avons :

$$A = \{ D_{P1}, D_{G1} \} = \{ \{a, b\}, \{a, e\} \}$$

$$\mathcal{D}_P^A = \{ D_{P1} \} = \{ \{a, b\} \}$$

$$\overline{\mathcal{D}_P^A} = \mathcal{D}_P - \mathcal{D}_P^A = \{ D_{P2} \} = \{ \{c, d\} \}$$

Avec cet ensemble de dimensions, nous pouvons agréger les valeurs de P sur ces dimensions. Nous créons le groupe de processus P_{agg} tel que⁴ :

$$P_{agg} = \overline{\mathcal{D}_P^A} G_{\overline{\mathcal{D}_P^A}, \text{Sum}(V)}(P)$$

Dans notre exemple, nous obtenons :

P_{agg}	$D_{P_{agg}1}$	value
	a	$V_{ac} + V_{ad}$
	b	$V_{bc} + V_{bd}$

Il ne nous reste plus qu'à réaliser la Σ -jointure entre P_{agg} et G . Sur notre exemple, nous obtenons alors le groupe de processus suivant :

$P_{agg} \bowtie_{\Sigma} G$	D_{P1}	D_{P2}	value
	a	f	$V_{ac} + V_{ad} + V_{af}$
	a	g	$V_{ac} + V_{ad} + V_{ag}$

4. $G_{\text{Sum}()}$ correspond à la fonction d'agrégation somme

5.1.3 Implémentation en SQL

Nous présentons dans cette section l'application en SQL de l'approche déclarative basée sur l'algèbre relationnelle. Nous commençons par décrire les algorithmes correspondant à la Π -jointure, au calcul de l'ensemble d'appariement et à la Σ -jointure. Puis nous présentons une variation sur cette application en SQL.

La Σ -jointure et la Π -jointure sont aisément traduites en SQL. Par exemple, en considérant les groupes G_{p1} et G_c présentés au début de ce chapitre, l'opération $G_{p1} \bowtie_{\Pi} G_c$ se traduit en SQL par la requête⁵ :

```
SELECT Gp1.Dp11, Gc.Dc1, Gc.Dc2, Gp1.value * Gc.value
FROM Gp1, Gc
WHERE Gp1.Dp11 = Gc.Dc1;
```

Le calcul des conditions d'appariement utilisées pour ces deux opérations ne peut, quant à lui, pas être réalisé par une simple requête SQL. Nous définissons une fonction qui retourne la condition d'appariement qui sera utilisée pour réaliser la (Σ ou Π)-jointure entre deux groupes.

Pour déterminer la condition d'appariement, $WHERE\ Gp1.Dp11=Gc.Dc1$ dans notre exemple, nous devons déterminer quelles sont les dimensions des deux groupes qui ont une intersection non nulle. Il est donc nécessaire de connaître la liste des dimensions des groupes et les mots-clefs qu'elles contiennent. Dans un SGBDR, ces informations sont disponibles dans les métadonnées sur les tables représentant les groupes.

Nous pouvons aussi rendre explicite ces métadonnées en utilisant deux nouvelles relations :

- `dimension(dimension_id, keyword)` décrit les domaines de définition des dimensions ;
- `groups_dimension(group_id, dimension_id)` décrit les dimensions composant les groupes.

Sur l'exemple présenté au début de ce chapitre, le contenu de ces relations pour stocker les méta données des trois groupes est :

5. Dans un SGBDR, le groupe G_{p1} est stocké dans une table G_{p1} et sa dimension D_{p11} correspond à une colonne $Dp11$. La sérialisation du groupe G_c reprend le même principe.

dimension	dimension_id	keyword
	Dp11	Truck
	Dp11	Barge
	Dp21	Oil
	Dp21	Coal
	Dc1	Truck
	Dc1	Barge
	Dc2	Oil
	Dc2	Coal
groups_dimension	group_id	dimension_id
	gp1	Dp11
	gp2	Dp21
	gc	Dc1
	gc	Dc2

La chaîne de caractères correspondant à la condition d'appariement entre deux groupes est construite en parcourant leurs dimensions. Si deux dimensions ont une intersection non nulle, nous les ajoutons à la condition. Le listing suivant présente l'algorithme utilisant des requêtes SQL sur les tables `dimension` et `groups` pour déterminer la condition d'appariement entre deux groupes⁶ :

```
MatchingSet (group1Id,group2Id)
matchingSetQuote = '1'
For all d1_id in 'SELECT dimension_id
                FROM groups_dimension
                WHERE group_id=group1Id' do
For all d2_id in 'SELECT dimension_id
                FROM groups_dimension
                WHERE group_id=group2Id' do
If 'SELECT COUNT(dimension1.keyword)
   FROM dimension dimension1
   INNER JOIN dimension dimension2
   ON dimension1.dimension_id = d1_id
   AND dimension2.dimension_id = d2_id
   AND dimension1.keyword = dimension2.keyword' != 0
matchingSetQuote = CONCAT(matchingSetQuote,
                          ' AND ', d1_id, ' = ', d2_id)
Endif
Endforall
Endforall
return matchingSetQuote
```

Il y a cependant deux problèmes avec cette implémentation :

6. Nous disposons d'une fonction `CONCAT()` qui concatène des chaînes de caractères.

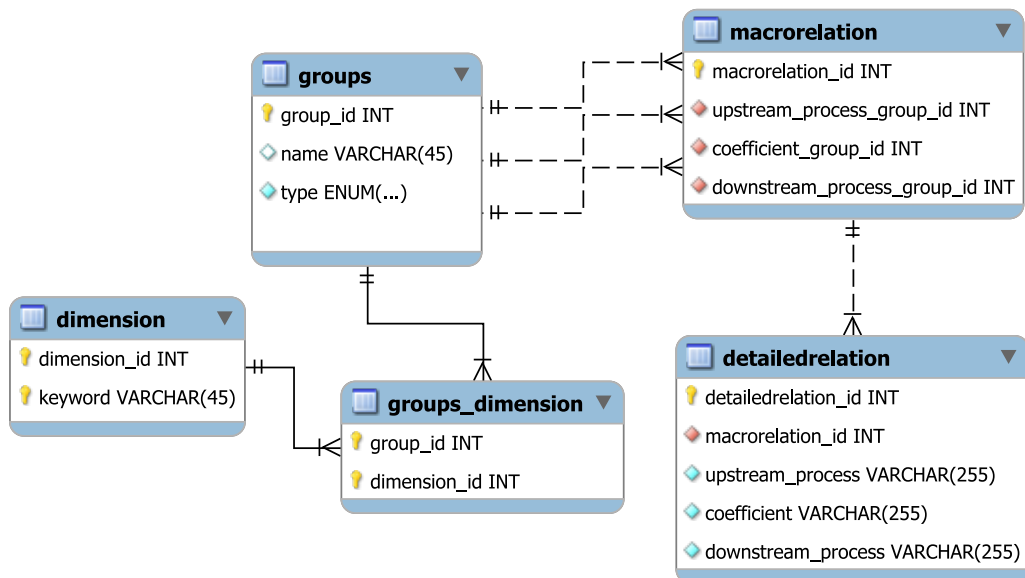


FIGURE 29: Modèle Conceptuel de Données de l'implémentation dans un SGBDR.

- nous ne créons pas de relations détaillées, *i.e.* : d'arcs dans le graphe détaillé, mais nous calculons directement les impacts ;
- les dimensions ne sont pas partagées entre les groupes, on ne peut donc pas modifier plusieurs groupes en modifiant une seule dimension.

De plus, si on a une table par groupe, nous accroissons inutilement le nombre de tables de la base de données. Chaque groupe contient son propre sous-ensemble de dimensions. Chaque table contient donc chaque coordonnée du groupe qu'elle représente. Le nombre de tuples d'une table d'un groupe correspond à la taille du produit cartésien des dimensions alors que les groupes ne sont que des pointeurs vers des processus et des coefficients.

C'est pourquoi, l'implémentation SQL de cette approche repose sur le schéma de base de données présenté dans la figure 29, il comprend :

- une table pour les groupes ;
- une table pour les dimensions détaillant quels sont les mots-clefs des dimensions ;
- une table pour l'association groupe - dimension ;
- une table pour les macro-relations qui précise quel est le groupe amont, le groupe de coefficients et le groupe aval ;
- une table pour les relations détaillées qui précise quel est le processus amont, le coefficient et le processus aval.

Au lieu de créer une table par groupe, l'utilisateur décrit uniquement les groupes (son nom et son type, ses dimensions, les mots-clefs de ses dimensions). Pour convertir il faut ensuite créer des macro-relations qui seront converties en relations détaillées directement. Nous convertissons donc des tuples contenus dans la table des macro-relations en tuples dans la table des relations détaillées.

Avec cette implémentation, les étapes de conversion d'une macro-relation sont :

1. Création des tables temporaires correspondant aux groupes impliqués
2. Calcul de la Π -jointure et création du groupe résultant
3. Calcul de la Σ -jointure et insertion des relations détaillées
4. Suppression des tables temporaires des trois groupes

Nous associons à chaque coordonnée d'une cellule des groupes l'adresse de l'élément qui est référencé. Le groupe résultant de la Π -jointure possède un attribut pour l'adresse du processus et un attribut pour celle du coefficient. Lors du calcul de la σ -jointure nous insérons un tuple dans la table des relations détaillées en utilisant ces adresses.

Cette implémentation requiert, en plus de la procédure de calcul de la condition d'appariement, une procédure pour la création des groupes temporaires et une procédure pour la conversion d'une macro-relation.

Pour simplifier l'utilisation de cette implémentation, nous pouvons appeler la procédure de conversion d'une macro-relation à chaque modification de la table représentant les macro-relations (à l'aide d'un *trigger* post-insertion sur la table macro-relation décrite dans le schéma de la Figure 29). Si une macro-relation existante est modifiée, il faut d'abord supprimer les relations déjà créées à partir de cette macro-relation. Puis, dans tous les cas, nous créons les relations détaillées. L'unicité des relations détaillées est garantie par la clef primaire de la table des relations détaillées qui contient le processus amont, le coefficient et le processus aval.

5.1.4 Expérimentation sur le passage à l'échelle de l'implémentation en SQL

Nous avons réalisé des tests de passage à l'échelle de cette implémentation en SQL pour comparer son efficacité en temps de calcul par rapport à l'algorithme procédural présenté dans le chapitre 3. Cette expérimentation reprend les mêmes conditions que celles portant sur l'algorithme procédural : jeu de données synthétiques, groupes simplifiés de 10 000 processus maximum et PC identique.

La première expérience a été réalisée pour comparer l'implémentation SQL avec la version naïve et la version optimisée de l'algorithme de conversion procédural. Nous avons généré un groupe de processus amont avec un nombre croissant de mots-clés contenus dans une seule dimension. Les coordonnées des cellules de ce groupe ne comportant qu'un seul mot-clé, nous avons autant de processus que de mot-clés dans la dimension. Nous ne créons que des relations normales : les groupes de coefficients et de processus avals sont des copies du groupe de processus amont. Il y a donc autant de relations détaillées créées que de processus dans le groupe amont.

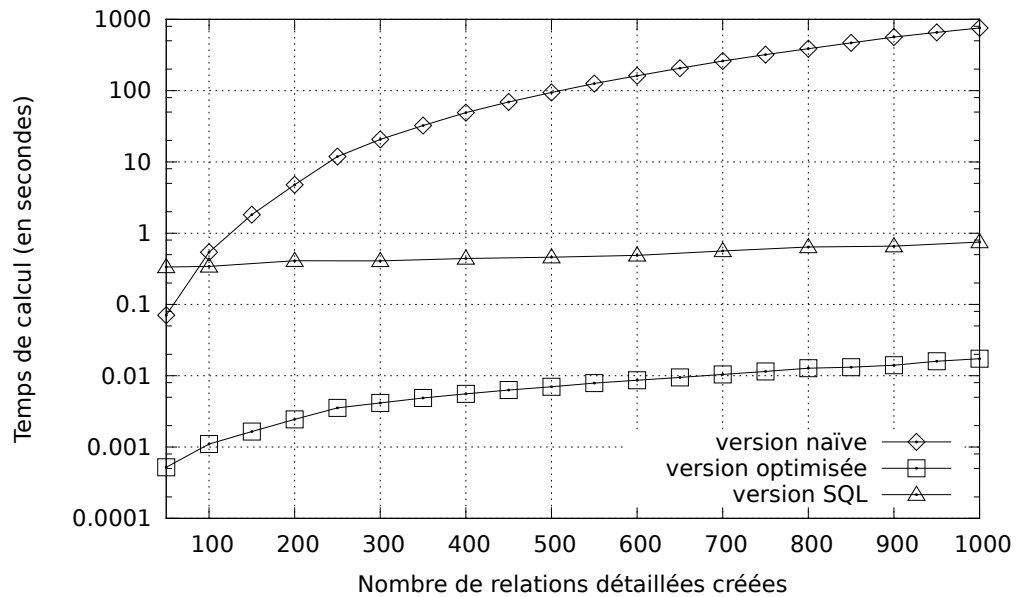


FIGURE 30: Comparaison du temps de calcul pour convertir des macro-relations de taille croissantes entre la version SQL et les versions procédurales naïve et optimisée de l'algorithme de conversion des macro-relations. L'échelle de l'axe des ordonnées est logarithmique.

Dans le graphique présenté figure 30, nous avons tracé le temps de calcul nécessaire aux deux versions de l'algorithme procédural et à l'implémentation SQL en fonction du nombre de processus dans le groupe amont. Nous avons limité le nombre de processus amonts à 1 000 car le temps de conversion des macro-relations pour la version naïve devenait trop important. Nous constatons que la version SQL est moins efficace que les deux versions procédurales pour des groupes dont la taille est inférieure à 100 processus, mais est plus efficace que la version naïve pour des groupes de plus grande taille. Cependant, cette version est dans tous les cas moins efficace que la version optimisée de l'algorithme procédural.

La deuxième expérience a été réalisée pour comparer l'implémentation SQL avec la version optimisée de l'algorithme de conversion procédural pour des groupes de plus grande taille. Le résultat de cette expérience est présenté dans la figure 31. Nous constatons qu'il y a au minimum deux ordres de grandeurs de plus pour la version SQL. Cette différence s'explique probablement par la surcharge de travail nécessaire au fonctionnement d'un SGBDR, tel que l'interprétation d'une requête SQL.

5.1.5 Limites de l'approche basée sur l'algèbre relationnelle

Le *trigger* mis en place sur la table des macro-relations n'est pas suffisant. Les mises à jour du contenu de la base nécessitant de re-converter une macro-relation peuvent aussi avoir lieu sur la définition des dimensions des groupes impliqués dans une macro-relation et sur l'ontologie des mots-clés. Il est donc nécessaire d'observer les changements intervenant sur le contenu des tables en

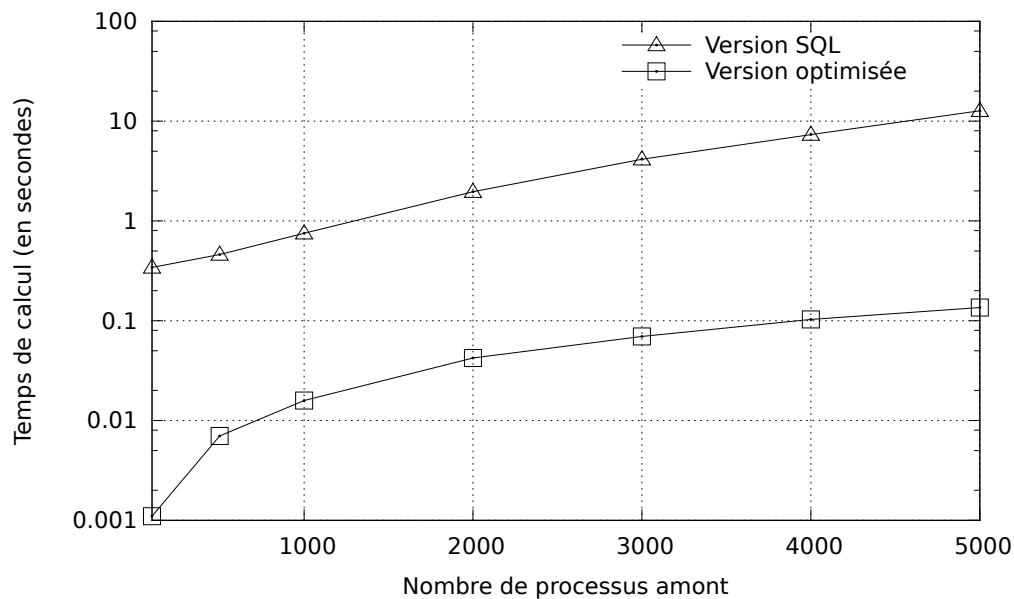


FIGURE 31: Comparaison du temps de calcul pour convertir des macro-relations de taille croissantes entre la version SQL et la version procédurale optimisée de l'algorithme de conversion des macro-relations. L'échelle de l'axe des ordonnées est logarithmique.

amont des macros-relations : la table des dimensions et la table de l'association groupe - dimension. Mais il est aussi nécessaire de surveiller l'ontologie des mots-clefs. Hors cette ontologie est plus aisément représenté en utilisant des technologies basées sur la logique de description à la place de l'algèbre relationnelle.

A l'aide de l'ontologie des mots-clefs, nous pouvons répondre à des requêtes sur le graphe détaillé liées à la sémantique des processus. Par exemple, une requête du type : « Quels sont les flux cumulés des processus indexés par un mot-clef sur un processus donné ? ». Pour répondre à une telle requête, il faut parcourir tout les processus amonts d'un processus donné dans le graphe détaillé. Ceci requiert de réaliser une requête récursive sur le contenu des relations détaillées. Ce type de requêtes est possible en SQL avec les *recursive common table expressions* introduites dans le standard SQL :1999 [55]. Mais comme l'ontologie des mots-clefs est stockée en utilisant des outils du web sémantique et pour éviter de recourir à deux technologies différentes, nous avons choisi de réaliser ce type de requêtes avec le langage de requête SPARQL comme nous le présentons dans la section suivante.

5.2 APPROCHE BASÉE SUR LES STANDARDS DU WEB SÉMANTIQUE

Nous nous intéressons dans cette section à répondre à des requêtes sur la composition des processus. Plus précisément, nous nous intéressons à des requêtes portant sur l'analyse des processus amonts d'un processus donné. Ce type de requête implique de s'intéresser aux relations de composition amont d'un processus, donc à parcourir tous les chemins dans le graphe jusqu'à un

noeud donné, et à exploiter l'ontologie des mots-clefs.

Avec l'ontologie des mots-clefs, nous pouvons répondre à des requêtes sur la composition des processus portant sur leurs sémantiques. Nous pouvons, entre autres, nous intéresser à des requêtes permettant de connaître l'impact d'un ensemble de processus sur un processus précis. Pour répondre à ce type de requête, nous avons besoin à la fois d'analyser les relations de composition amont d'un processus et les relations entre les mots-clefs. Il faut parcourir tous les processus amonts et ne retenir que ceux qui respectent certains critères précisés dans la requête. L'intérêt repose sur l'expression de critères mettant à profit l'indexation sémantique des processus.

Nous souhaitons, par exemple, connaître l'impact des processus de transport sur un processus de production d'électricité. Pour répondre à ce questionnement, nous devons récupérer l'ensemble des processus qui sont indexés par des mots-clefs en relation avec un mot-clef décrivant les modes de transport et calculer leurs flux cumulés pour le processus de production d'électricité.

Les différents standards du web sémantique ont été conçus pour représenter et manipuler des graphes orientés. Aussi bien l'ontologie des mots-clefs que le graphe détaillé des processus sont des graphes orientés. Les outils basés sur ces standards sont donc parfaitement adaptés pour exprimer des requêtes portant sur l'analyse de ces graphes et y répondre.

Dans un premier temps, nous présentons une modélisation OWL de l'ontologie des mots-clefs puis une modélisation OWL du graphe détaillé. Nous présentons ensuite comment répondre à des requêtes sémantiques portant sur la composition des processus en utilisant le langage de requêtes SPARQL.

5.2.1 Modélisation OWL de l'ontologie des mots-clefs

Nous définissons une classe OWL `Keyword` pour représenter les mots-clefs. Chaque mot-clef sera donc une instance de cette classe. Nous définissons la propriété `isLinkedTo`, dont le domaine de définition et l'image sont des mots-clefs, qui définit la notion générique de prédicat entre deux mots-clefs. Chaque prédicat est une sous-propriété de `isLinkedTo`. La Figure 32 contient le graphe RDF correspondant à la définition en OWL de la classe `Keyword` et de deux prédicats entre des mots-clefs : le prédicat `subsumes` et le prédicat `uses`. La Figure 33 contient des instances de la classe `Keyword` correspondant à l'ontologie des mots-clefs des différents modes de transport.

Dans cet exemple, nous avons exprimé une relation de subsomption entre certains mots-clefs à l'aide du prédicat `isSubsumedBy`. Comme nous avons défini que les prédicats `subsumes` et `isSubsumedBy` sont l'inverse l'un de l'autre, un moteur d'inférence nous permet d'obtenir de nouveaux faits, tels que⁷ :

7. Tous les exemples RDF sont exprimés dans la syntaxe turtle [5], nous utilisons un espace de nommage XML vide pour présenter les concepts de notre approche.

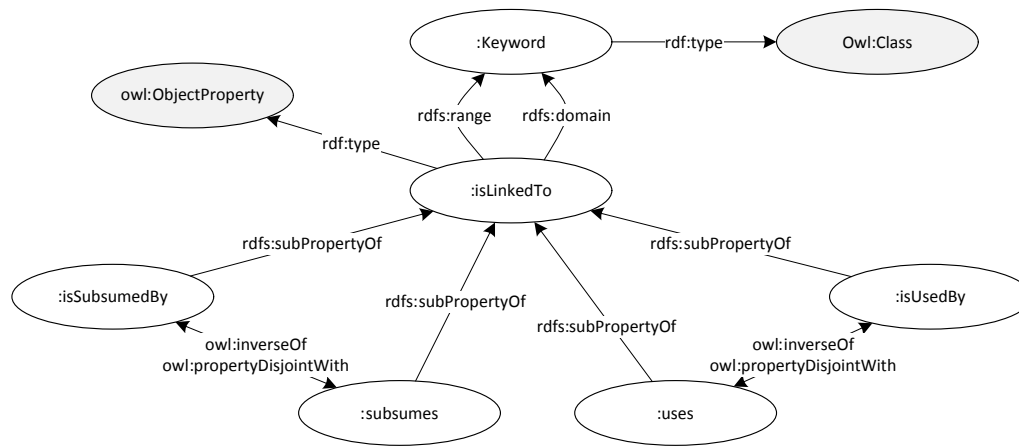


FIGURE 32: Graphe RDF du vocabulaire de l'ontologie des mots-clés.

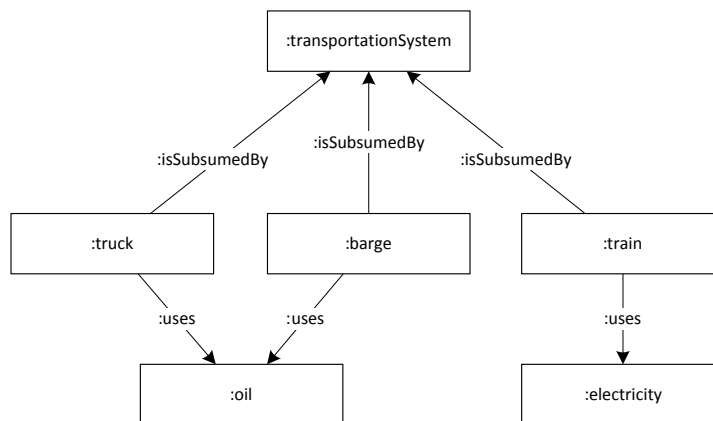


FIGURE 33: Graphe RDF de l'ontologie des mots-clés correspondant aux modes de transport.

```
:transportationSystem :subsumes :barge, :train, :truck .
```

Avec cette modélisation, nous pouvons récupérer des ensembles de mots-clés, pour les dimensions par exemple, *via* des requêtes SPARQL [73] sur notre ontologie. Sur le graphe RDF de la Figure 33⁸, la requête permettant de récupérer tous les mots-clés décrivant des modes de transport utilisant du pétrole est :

```
SELECT ?keyword
WHERE
{
  ?keyword :isSubsumedBy :transportationSystem .
  ?keyword :uses :oil .
}
```

Le résultat de cette requête contient les nœuds RDF truck et barge.

Dans sa révision 1.1, le standard SPARQL introduit la notion de *property-path* qui permet de trouver les chemins entre deux nœuds d'un graphe RDF. Nous

8. Dans nos figures représentant des extraits de graphes OWL, nous représentons les classes par des ovales et les instances de ces classes par des rectangles.

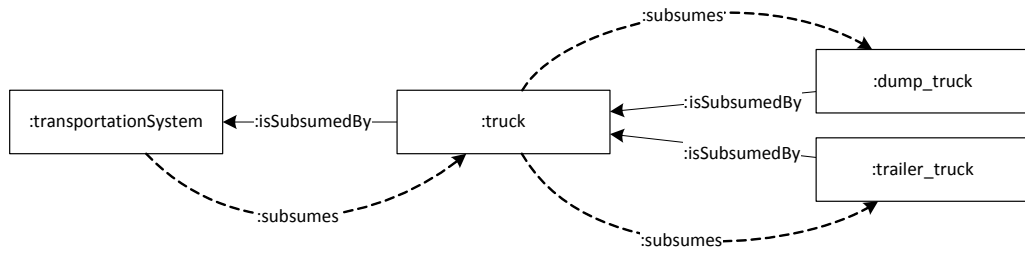


FIGURE 34: Graphe RDF de l'ontologie des mots-clefs correspondant à différents types de camions. Les arcs en pointillés représentent les propriétés inverses inférées par un raisonneur.

pouvons utiliser cet opérateur pour déterminer s'il existe des chemins de propriétés de taille variable dans l'ontologie des mots-clefs. Cette notion peut nous être utile pour récupérer des mots-clefs qui sont liés au travers de plusieurs mots-clefs intermédiaires.

Pour illustrer son intérêt, nous reprenons le graphe RDF de la Figure 33 et nous ajoutons des mots-clefs plus spécifiques que truck tel qu'illustré dans la Figure 34. La requête suivante permet de récupérer les mots-clefs qui ont une chaîne de propriétés jusqu'au mot-clef transportSystem uniquement composé de la propriété subsumes⁹ :

```
SELECT ?keyword
WHERE
{
    :transportationSystem :subsumes+ ?keyword .
}
```

Le résultat de cette requête appliquée au graphe RDF de la Figure 34 contient les nœuds RDF truck, dump_truck et trailer_truck.

5.2.2 Modélisation OWL du graphe détaillé

Pour modéliser les processus et les coefficients nous définissons une classe Element qui permet de factoriser leurs deux caractéristiques communes : l'indexation avec des mots-clefs et une valeur numérique¹⁰. Le graphe RDF correspondant à cette modélisation est présenté dans la Figure 35.

Les classes Process et Coefficient sont des sous-classes de la classe Element. Les propriétés hasValue et hasKeyword ont pour domaine de définition des instances de la classe Element. hasValue a pour image un nombre décimal et hasKeyword a pour image une instance de la classe Keyword.

9. Nous avons choisi d'illustrer cet opérateur après l'utilisation d'un raisonneur pour exploiter les liens de subsumption inverses mais nous aurions aussi pu utiliser la syntaxe `^:isSubsumedBy+` pour considérer les chaînes de propriétés inverses.

10. Un processus est normalement associé à un ensemble de flux vers l'environnement, mais par soucis de simplification, nous présentons une modélisation OWL dans laquelle un processus n'est associé qu'à un à un seul flux de nature indifférenciée.

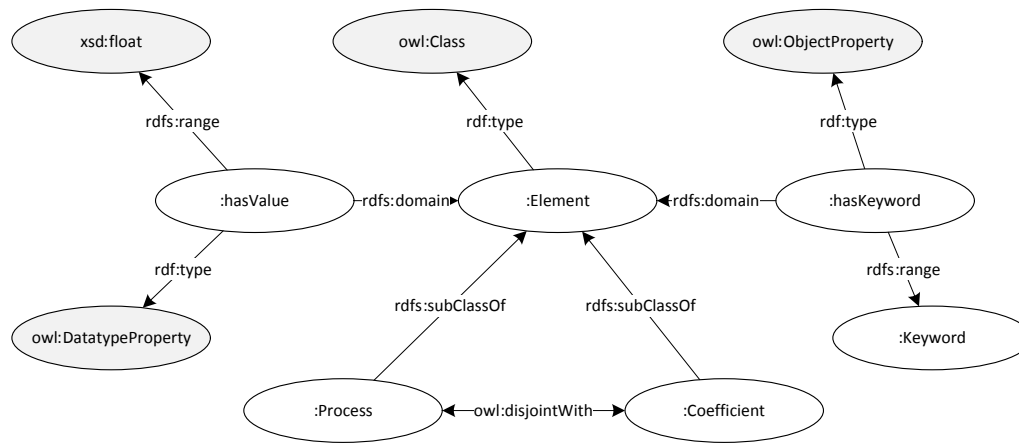
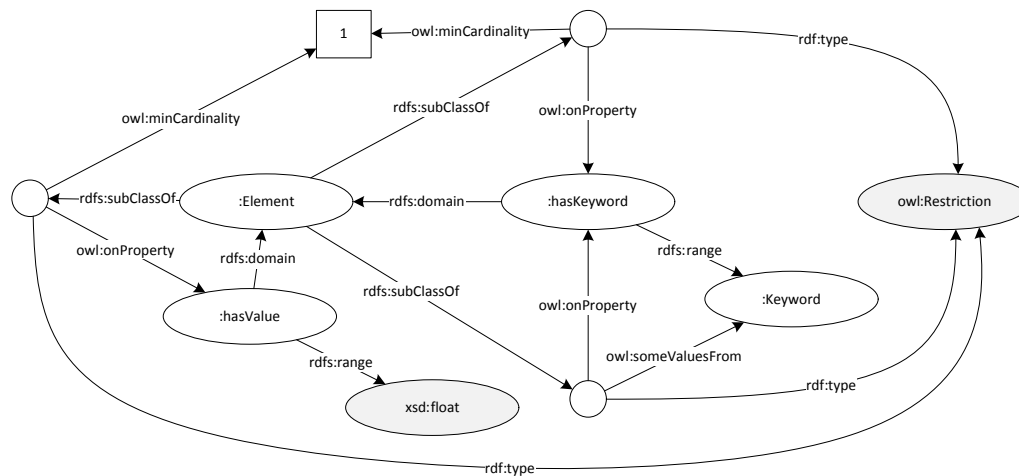


FIGURE 35: Graphe RDF des classes OWL des processus et coefficients.

FIGURE 36: Graphe RDF des restrictions sur la classe Element. Les nœuds sans libellé représentent des *blank nodes* RDF.

Nous mettons en place deux restrictions sur la classe Element :

- un élément doit être indexé par au moins un mot-clé ;
- un élément a exactement une valeur.

Ces restrictions sont présentées dans la Figure 36.

Les propriétés OWL n'étant que binaires, nous ne pouvons pas associer deux processus entre eux en précisant quel est le coefficient qui pondère cette relation si nous n'utilisons qu'une propriété pour une relation détaillée. Dans le standard RDF [49], il est possible d'utiliser le mécanisme de réification d'une assertion pour définir des relations n-aires. Nous avons cependant choisi de respecter les bonnes pratiques de réification des relations n-aires en OWL du W3C. La recommandation sur la modélisation des relations n-aires en OWL

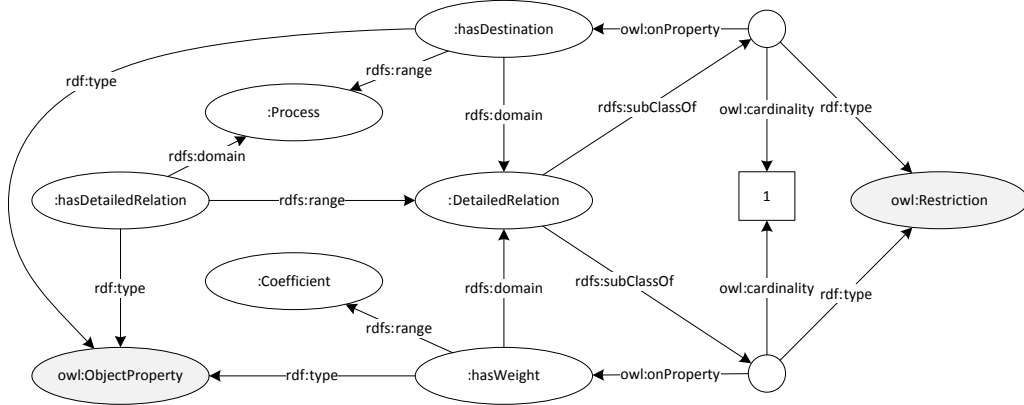


FIGURE 37: Graphe RDF de la réification des relations détaillées et ses restrictions.

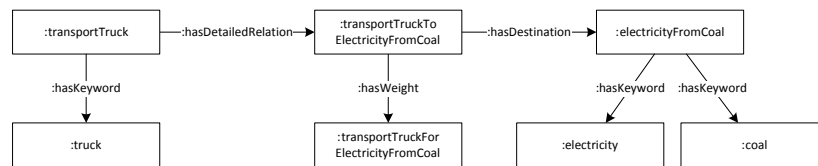


FIGURE 38: Graphe RDF des relations détaillées entre le processus de transport de marchandises par camion et le processus de production d'électricité à partir de charbon.

[40] préconise de ne pas avoir recours à ce mécanisme¹¹.

Pour modéliser les relations détaillées, nous réifions la relation entre deux processus pour préciser le coefficient qui la pondère. Nous ajoutons une classe DetailedRelation qui est associée à un processus en utilisant la propriété hasDestination et à un coefficient en utilisant la propriété hasWeight. Un processus est lié à une relation détaillée à l'aide de la propriété hasDetailedRelation. Le graphe RDF de cette classe et de ses restrictions est présenté dans la Figure 37.

La Figure 38 présente les instances correspondant à une relation détaillée entre le processus de transport de marchandises par camion, transportTruck, et le processus de production d'électricité à partir de charbon, electricityFromCoal. Cette relation est pondérée par le coefficient transportTruckForElectricityFromCoal. Le processus transportTruck est indexé par le mot-clé truck et le processus electricityFromCoal est indexé par les mots-clés electricity et coal.

11. Comme le précise cette recommandation, la réification RDF permet d'ajouter des métadonnées sur le triplet RDF lui-même plutôt que sur la relation entre les deux entités impliquées dans la relation. Ce mécanisme n'a pas le même sens que d'ajouter des métadonnées sur la relation elle-même. La réification RDF est plus adéquate pour préciser, par exemple, des métadonnées sur la création d'une assertion (comme sa date d'ajout au graphe RDF).

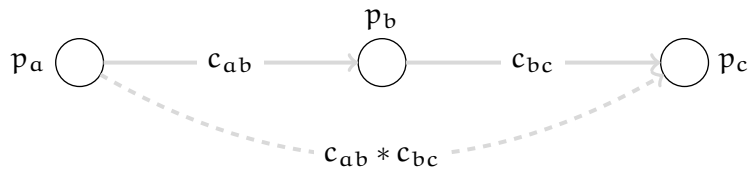


FIGURE 39: Graphe détaillé et sa fermeture transitive, l'arc en pointillé représente la fermeture transitive des deux relations détaillées $((p_a, p_b), c_{ab})$ et $((p_b, p_c), c_{bc})$.

5.2.3 Requêtes sémantiques et graphe détaillé inverse

Pour répondre à une requête sémantique sur la composition d'un processus, nous avons besoin de :

- trouver ses processus amonts qui sont indexés par un ou plusieurs mots-clefs, puis
- calculer les flux cumulés de ces processus pour le processus étudié.

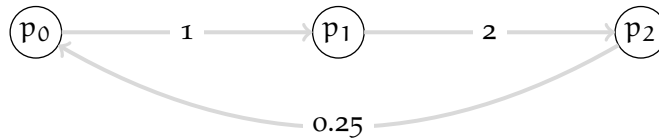
La condition sur l'indexation des processus amonts peut s'exprimer de la même façon que lors de la récupération d'un ensemble de mots-clefs pour exprimer une dimension. Pour calculer les flux cumulés nous devons connaître les différents chemins dans le graphe depuis un processus amont respectant la condition sur l'indexation jusqu'au processus étudié. Il faut ensuite sommer le produit des coefficients des différents chemins. En d'autres termes, il faut calculer la fermeture transitive du graphe détaillé.

La fermeture transitive du graphe détaillé est un graphe dans lequel il existe un arc entre deux processus s'il y a un chemin entre ces deux processus dans le graphe détaillé. Les pondérations de ces arcs seront égales au produit des coefficients du chemin. S'il existe plusieurs chemins entre deux processus dans le graphe détaillé, la pondération de l'arc entre ces deux processus dans la fermeture transitive sera égale à la somme du produit des coefficients.

Exemple : soit trois processus p_a , p_b et p_c et deux relations détaillées $((p_a, p_b), c_{ab})$ et $((p_b, p_c), c_{bc})$. La fermeture transitive du graphe détaillé contient la relation détaillée $((p_a, p_c), c_{ab} * c_{bc})$. Ce graphe détaillé et sa fermeture transitive sont représentés sur la figure 39.

Nous pouvons utiliser un raisonneur sur notre modèle en OWL pour calculer la fermeture transitive du graphe des relations détaillées. Mais la logique de description n'étant pas conçue pour réaliser des opérations arithmétiques, ce calcul peut se réaliser en utilisant un langage de règles tel que le langage SWRL [45] enrichi par les opérateurs de SQWRL [65] (qui fournit, entre autres, des opérateurs arithmétiques).

Mais, s'il y a un cycle dans le graphe détaillé, la fermeture transitive ne peut plus être calculé simplement en sommant le produit des coefficients des chemins dans le graphe. Nous devons alors calculer l'inverse de la matrice des

FIGURE 40: Graphe détaillé entre trois processus $p_i, i \in \{0, 1, 2\}$.

coefficients pour récupérer les valeurs des coefficients transitifs¹².

Les pondération du graphe correspondant à la fermeture transitive du graphe détaillé correspondent aux coefficients de la matrice inverse. Ce nouveau graphe est le graphe détaillé inverse. Ses coefficients nous permettent de calculer directement les flux cumulés pour un processus à partir des flux des autres processus. Le calcul du total des flux cumulés pour un processus se réduit donc à la somme du produit des flux des processus amonts par les coefficients inverses.

Exemple : soit trois processus $p_i, i \in [0, 2]$. On note $F(p_i)$ et $F_{\text{cumulés}}(p_i)$ respectivement les flux et les flux cumulés d'un processus p_i . Les relations détaillées entre ces trois processus sont présentées dans le graphe détaillé de la Figure 40.

Le système d'équation des coefficients de dépendances de ce graphe détaillé est :

$$\begin{cases} p_0 &= 0.25p_2 + F(p_0) \\ p_1 &= p_0 + F(p_1) \\ p_2 &= 2p_1 + F(p_2) \end{cases}$$

Ce système sous sa forme matricielle $Ax = b$ correspond à :

$$\underbrace{\begin{pmatrix} 1 & 0 & -0.25 \\ -1 & 2 & 0 \\ 0 & -2 & 1 \end{pmatrix}}_A \underbrace{\begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix}}_x = \underbrace{\begin{pmatrix} F(p_0) \\ F(p_1) \\ F(p_2) \end{pmatrix}}_b$$

La matrice inverse des coefficients est :

$$A^{-1} = \begin{pmatrix} 2 & 1 & 0.5 \\ 2 & 2 & 0.5 \\ 4 & 4 & 2 \end{pmatrix}$$

Le graphe détaillé inverse correspondant à cette matrice inverse est représenté dans la Figure 41.

Pour déterminer les flux cumulés d'un processus amont sur un processus aval, il suffit alors de multiplier le coefficient inverse entre ces deux processus

¹². Voir le chapitre [CALCUL DES FLUX CUMULÉS](#) pour plus de détails sur l'inversion de la matrice des coefficients.

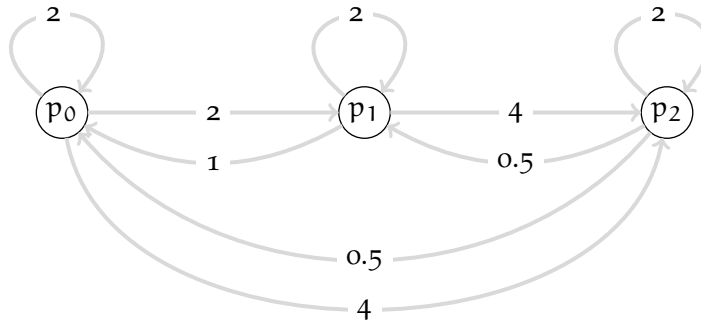
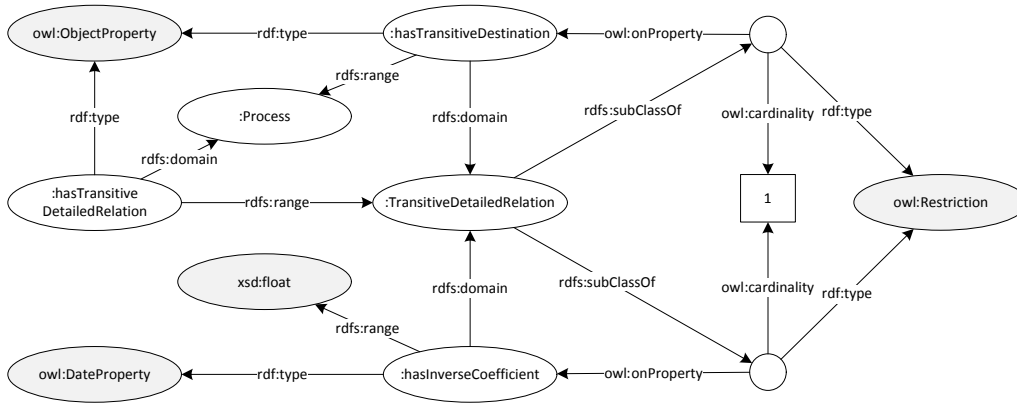
FIGURE 41: Graphe détaillé inverse entre trois processus p_i , $i \in [0, 2]$.

FIGURE 42: Graphe RDF de la réification des relations détaillées et ses restrictions.

par les flux du processus amont. Par exemple, les flux cumulés de p_0 pour le processus p_2 seront égales à $4 * F(p_0)$.

Nous ajoutons ensuite les coefficients de la matrice A^{-1} à la modélisation OWL en utilisant une nouvelle classe pour les relations détaillées transitives et de nouvelles propriétés pour décrire les processus et le coefficient inverse impliqués. La classe des relations détaillées transitives et les propriétés liées sont présentées dans la Figure 42. Les coefficients transitifs sont directement représentés sous la forme d'un nombre décimal à l'aide de la propriété de données `hasInverseCoefficient`.

Il est alors possible de répondre à une requête sur les amonts d'un processus se basant sur la sémantique des mots-clés en utilisant SPARQL. En considérant le graphe détaillé présenté dans la Figure 38, nous pouvons récupérer les flux cumulés des processus de transport amont pour le processus de production d'électricité à partir de charbon à l'aide de la requête suivante :

```
SELECT SUM(?flow) AS ?cumulatedFlows
WHERE
{
  ?process :hasTransitiveDetailedRelation :electricityFromCoal .
  ?process :hasValue ?flow .
```



```

    ?process :hasKeyword ?keyword .
    ?keyword ^:isSubsumedBy+ :transportationSystem
  }
GROUP BY ?process

```

L'inconvénient de cette approche est l'accroissement du nombre de triplets que nous devons stocker dans le graphe RDF. Si nous ajoutons à notre implémentation OWL tous les coefficients de la matrice inverse et que nous avons n processus, nous avons au pire des cas n^2 coefficients inverses (donc $3n^2$ triplets car il nous faut au minimum trois triplets RDF pour déclarer une relation détaillée transitive). Le pire des cas se présente quand nous n'avons qu'une seule composante fortement connexe dans le graphe détaillé.

5.3 SYNTHÈSE

Nous avons présenté dans ce chapitre deux approches déclaratives pour simplifier l'implémentation et l'utilisation de notre méthode. Pour l'approche basée sur l'algèbre relationnelle, nous avons introduit deux nouveaux opérateurs pour réaliser la multiplication d'un groupe de processus par un groupe de coefficients et l'addition de deux groupes de processus. Ils nous permettent de convertir des macro-relations en relations détaillées. Nous avons proposé une implémentation SQL de cette approche ainsi que le schéma de base de données sur lequel elle se base. L'approche basée sur les outils du web sémantique que nous avons présentée permet de modéliser et d'interroger le graphe détaillé en utilisant les informations contenues dans l'ontologie des mots-clefs.

CALCUL DES FLUX CUMULÉS

Dans la méthodologie de l'ACV, l'environnement est découpé en plusieurs compartiments. Nous avons, par exemple, un compartiment pour le CO₂ présent dans l'atmosphère, un autre pour les cours d'eaux, etc.. Chaque processus interagit avec ces compartiments : émissions de gaz à effet de serre vers l'atmosphère, rejets d'eaux usées dans les cours d'eau... Ces interactions sont nommées *flux élémentaires*. Les processus étant interdépendants, l'ensemble des interactions d'un processus avec l'environnement doit prendre en compte aussi bien ses flux élémentaires que les flux élémentaires de ses prédécesseurs, on parle alors de *flux cumulés* d'un processus.

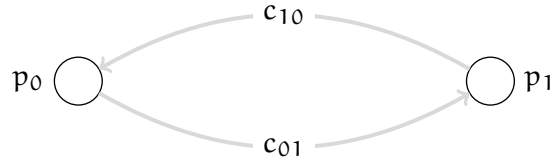
Dans ce chapitre nous présentons la méthode de calcul des flux cumulés des processus d'un point de vue mathématique puis nous présentons sa réalisation effective d'un point de vue algorithmique. Nous présentons également une condition essentielle portant sur sa faisabilité. Pour finir, nous nous proposons d'appliquer une méthode de décomposition du graphe des relations détaillées pour obtenir une information plus précise quant à la faisabilité du calcul des flux cumulés. Nous proposons ensuite une méthode d'optimisation basée sur cette décomposition qui permet de limiter les calculs à réaliser dans le cas d'une modélisation dynamique. Pour simplifier la compréhension du chapitre, nous supposons que l'environnement ne possède qu'un seul compartiment. Un processus n'est donc associé qu'à un seul flux élémentaire et nous ne calculons qu'un seul flux cumulé par processus.

6.1 CALCUL MATRICIEL DE L'ANALYSE EN CYCLE DE VIE

Nous supposons que nous avons n processus numérotés de 1 à n . Pour tout $(i, j) \in \{1, \dots, n\}^2$, on note c_{ij} le coefficient de dépendance entre le processus i et le processus j . Un coefficient non nul c_{ij} signifie que la réalisation d'une unité du processus j nécessite c_{ij} unité du processus i ¹.

Ces coefficients, aussi nommés *flux économiques*, ne représentent que les relations de dépendance directes (de proche en proche) entre processus. Pour déterminer le flux vers l'environnement d'un processus, nous devons prendre en compte son flux élémentaire et les flux indirects issus des chaînes de composition des processus. La somme du flux élémentaire d'un processus et des flux élémentaires de ses prédécesseurs pondérés par les coefficients de dépendance nous permet de déterminer son flux cumulé.

1. Les coefficients de dépendance sont ici normalisés. On considère que la réalisation d'un processus n'engendre qu'une unité du processus. Les processus de l'analyse en cycle de vie ne sont pas forcément normalisés, notamment dans les bases d'inventaires. Une des premières étapes de la méthode usuelle consiste à normaliser les processus.

FIGURE 43: Graphe de dépendance entre deux processus p_0 et p_1 .

On note F_e^i le flux élémentaire du processus i vers l'environnement et on note F_c^i son flux cumulé qui est égal à :

$$F_c^i = F_e^i + \sum_{j=0}^n (c_{ji} F_c^j)$$

La solution la plus évidente serait de parcourir l'ensemble des processus selon leur ordre de dépendance et de calculer leurs flux cumulés en fonction des processus précédents. Mais cette méthode n'est pas utilisable si nous avons un cycle dans le graphe de dépendance des processus. Nous utilisons donc une approche matricielle d'un inventaire.

Soit C la matrice carrée d'ordre n des coefficients de dépendance entre processus. On note $F_e = (F_e^1, \dots, F_e^n)$ le vecteur des flux élémentaires et $F_c = (F_c^1, \dots, F_c^n)$ le vecteur des flux cumulés. On a alors :

$$F_c = F_e + CF_e + C^2 F_e + \dots = \left(\sum_{k=0}^{+\infty} C^k \right) F_e$$

Les relations de dépendance entre les processus peuvent aussi être modélisées sous la forme d'un système d'équation linéaire tel que :

$$\begin{aligned} F_c &= CF_c + F_e \\ \Leftrightarrow F_c &= (I - C)^{-1} F_e \end{aligned}$$

Avec I correspondant à la matrice identité d'ordre n .

L'inversion de la matrice $(I - C)$ n'est possible que si elle n'est pas singulière. Nous pouvons nous en assurer en étudiant les puissances successives de C .

6.1.1 Faisabilité du calcul des flux cumulés

Considérons un cas simple de dépendance entre deux processus p_0 et p_1 qui dépendent l'un de l'autre, tel qu'illustré dans le graphe de la Figure 43, avec :

$$\begin{cases} F_c^0 = c_{10} F_c^1 + F_e^0 \\ F_c^1 = c_{01} F_c^0 + F_e^1 \end{cases} \Leftrightarrow \begin{cases} F_c^0 = c_{10} c_{01} F_c^0 + c_{10} F_e^1 + F_e^0 \\ F_c^1 = c_{01} c_{10} F_c^1 + c_{01} F_e^0 + F_e^1 \end{cases}$$

Le calcul des flux cumulés de ces deux processus ne peut se faire que si on a $c_{10} c_{01} < 1$. Dans le cas où on aurait $c_{10} c_{01} \geq 1$ leurs flux cumulés seraient

infinis, ce qui n'a physiquement pas de sens.

Plus généralement, pour obtenir une solution au calcul des flux cumulés, il est nécessaire et suffisant que la somme des puissances successives de C converge. Il faut donc que $\lim_{k=0}^{+\infty} C^k = 0$. Le terme 0 correspondant ici à une matrice carrée nulle d'ordre n . Cette convergence est assurée si le rayon spectral de C , que l'on note $\rho(C)$, est inférieur à 1 [98, 17]. Le rayon spectral de C correspond à :

$$\rho(C) = \{\max(|\lambda|) \text{ pour toutes les valeurs propres } \lambda \text{ de } C\}$$

Cette condition nous assure aussi que la matrice $(I - A)$ est inversible. La preuve suivante est une adaptation de la preuve présentée dans [51].

Démonstration. Nous cherchons à démontrer que le système $(I - C)x = 0$ a pour unique solution $x = 0$. Cette proposition est suffisante pour assurer l'inversibilité de la matrice $(I - C)$. On a :

$$(I - C)x = 0 \Leftrightarrow x = Cx \Rightarrow \|x\| = \|Cx\|$$

Par définition de la norme matricielle on a :

$$\|Cx\| \leq \|C\| \times \|x\|$$

Donc :

$$\|x\| \leq \|C\| \times \|x\| \Rightarrow (\|I\| - \|C\|)\|x\| \leq 0$$

Par définition on a $\rho(C) = \|C\|$ et $\|I\| = 1$. Comme $\rho(C) < 1$ on a $\|C\| < 1$, donc :

$$(\|I\| - \|C\|) = (1 - \|C\|) > 0$$

Donc :

$$(\|I\| - \|C\|)\|x\| \leq 0 \Rightarrow \|x\| = 0 \Rightarrow x = 0$$

Donc $(I - C)$ est inversible. □

D'un point de vue physique, une modélisation des interactions entre processus respectant cette condition nous assure que chaque processus ne consomme pas plus que ce qu'il ne produit.

6.1.2 Réalisation effective du calcul des flux cumulés

EN L'ABSENCE DE CYCLES DANS LA MODÉLISATION, nous pouvons utiliser plusieurs méthodes dont la complexité dans le pire des cas est au minimum en $O(n^2)$.

Le calcul des flux cumulés peut se réaliser en calculant les puissances successives de C jusqu'à la taille du chemin le plus long dans le graphe de dépendance que nous notons t_{\max} . Pour obtenir ces puissances successives, nous

devons utiliser des algorithmes de multiplication de matrices qui ont une complexité supérieure à $O(n^2)$ [16]².

Comme nous ne souhaitons obtenir que la valeur des flux cumulés, nous pouvons nous contenter de multiplier le vecteur F_d des flux directs t_{\max} fois par C . Dans sa version naïve, la multiplication d'un vecteur par une matrice a une complexité en $O(n^2)$ ³. Le pire des cas est atteint quand la taille du chemin le plus grand est égal au nombre de nœuds du graphe, cette méthode a donc une complexité en $O(n^3)$. Mais si $t_{\max} \ll n$, la complexité reste en $O(n^2)$.

La matrice des coefficients étant creuse, nous pouvons utiliser des algorithmes de produit de matrice creuse par des vecteurs plus optimisés qui ont une complexité en $O(\alpha nnz)$ avec nnz correspondant au nombre d'éléments de la matrice différents de zéro et α étant un coefficient réel très grand [75, 38]. Cette optimisation tient essentiellement à la structure de données utilisée pour stocker une matrice creuse [21]. L'utilisation d'algorithmes optimisés pour les matrices creuses n'a donc d'intérêt que pour des matrices suffisamment grandes, ce qui est le cas des matrices de coefficients de l'ACV.

Nous pouvons aussi appliquer un algorithme de substitution arrière ou avant [99] après avoir triangularisé la matrice des coefficients. Cette méthode nécessite de parcourir chaque ligne de la matrice C et chaque coefficient non nul de ces lignes. La complexité est donc ici aussi en $O(n^2)$.

EN PRÉSENCE DE CYCLES, et si la condition portant sur la faisabilité du calcul est respecté, nous pouvons utiliser une méthode de calcul directe ou itérative [75, 33]. Nous pouvons calculer la limite de la suite des puissances successives de C de manière itérative, calculer $(I - C)^{-1}$ ou résoudre le système d'équation $(I - C)x = b$. Les méthodes de résolution directes ont l'avantage de fournir une solution exacte aux valeurs des flux cumulés mais ont une complexité plus élevée que des méthodes de résolution itératives.

Pour une matrice quelconque, l'algorithme de résolution direct le plus courant est basé sur une décomposition LUP [16]. Cette technique décompose le système en deux matrices : une matrice triangulaire supérieure U , une matrice triangulaire inférieure L et une matrice de permutation P pour obtenir L et U . Il faut ensuite appliquer une méthode de substitution avant sur la matrice L et arrière sur la matrice U pour résoudre le système. Cet algorithme a une complexité en $O(n^3)$ avec n étant égal au nombre de coefficients dans la matrice des flux économiques. Il est évidemment possible d'utiliser des méthodes plus optimisées basées sur l'algorithme de Strassens par exemple [90]. La méthode

2. Une version naïve serait en $O(n^3)$. Les deux algorithmes les plus connus qui réussissent à diminuer le nombre d'opérations intuitivement nécessaires sont l'algorithme de Strassens qui a une complexité en $O(n^{2,81})$ [82] et l'algorithme de Coppersmith et Winograd qui a une complexité en $O(n^{2,38})$ [15].

3. La complexité du produit vecteur-matrice est $O(n.m)$ avec m étant la taille du vecteur. Dans notre cas le vecteur des flux directs étant de taille n , nous avons une complexité en $O(n^2)$.

LUP peut aussi être utilisée pour calculer l'inverse d'une matrice et a donc la même complexité [90].

Nous pouvons aussi calculer itérativement la valeur des flux cumulés en multipliant le vecteur des flux élémentaires F_e par la matrice des coefficients C et s'arrêter après un nombre d'itérations fixées au préalable ou dès que le module de la différence entre deux vecteurs F_c devient suffisamment faible⁴. Cette méthode est présentée dans l'algorithme suivant, l'opération $\|\cdot\|$ correspond à la norme euclidienne d'un vecteur :

Algorithme 5: Calcul itératif des flux cumulés

Entrées : $C, \text{iterationsMax}, \text{seuil}$

$i \leftarrow 0$

$F_c \leftarrow F_e$

$F_{c_{i-1}} \leftarrow F_c$

$\Delta \leftarrow \text{seuil}$

tant que $\Delta \geq \text{seuil}$ & $i < \text{iterationsMax}$ **faire**

$i++$

$F_{c_i} \leftarrow F_{c_{i-1}} * C$

$F_c \leftarrow F_c + F_{c_i}$

$\Delta \leftarrow \frac{\|F_{c_{i-1}} - F_{c_i}\|}{\|F_c\|}$

$F_{c_{i-1}} \leftarrow F_{c_i}$

retourner F_c

D'après les travaux présentés dans [67], moins d'une centaine d'itérations sont suffisantes pour obtenir une précision inférieure à l'incertitude pour la base d'inventaires ecoinvent [29]. La multiplication d'une matrice creuse par un vecteur se réalisant en $O(\alpha n n z)$ opérations, nous pouvons réaliser cette multiplication plus efficacement. Le nombre d'itérations nécessaires étant bien inférieur au nombre de processus d'une base d'inventaires, cette méthode a une complexité proche de $O(n^2)$. Elle est donc plus rapide que la méthode directe basée sur une décomposition LUP.

Cet algorithme s'applique aussi bien au calcul des flux cumulés dans le cas d'un graphe détaillé avec ou sans cycle. En l'absence de cycle, il suffit de contraindre le nombre d'itérations maximum à la taille du chemin le plus long du graphe détaillé.

En remplaçant la multiplication vecteur-matrice par une multiplication matrice-matrice, nous pouvons calculer les puissances successives de C . Et, étant donné que $\sum_{k=0}^{+\infty} C^k = (I - C)^{-1}$, nous pouvons, pour un nombre d'itération suffisant, obtenir une solution approchée de l'inverse de la matrice C . Mais dans ce cas-là, nous devons réaliser une opération de multiplication de matrices, avec, par

4. Les inventaires en cycle de vie sont entachés d'incertitudes qui peuvent permettre de déterminer un seuil de tolérance pour arrêter les itérations.

$$A = \begin{pmatrix} x & x & o & o \\ x & x & o & o \\ x & x & x & x \\ x & x & x & x \end{pmatrix} = \begin{pmatrix} A_1 & o \\ A_{21} & A_2 \end{pmatrix}$$

FIGURE 44: Exemple d'une matrice mise sous forme triangulaire supérieure par blocs.

exemple, l'algorithme de Strassens dont la complexité est en $O(n^{2,81})$ ⁵. Nous aurions donc une complexité équivalente à une méthode de résolution directe.

6.2 OPTIMISATION DU CALCUL

6.2.1 Décomposition par bloc de la matrice des flux économiques

Une matrice creuse peut-être mise sous la forme d'une matrice triangulaire supérieure ou inférieure par blocs. Cette forme permet d'optimiser un certain nombre d'opérations sur des matrices creuses, dont la résolution d'un système d'équation. Une matrice triangulaire par blocs supérieure, respectivement inférieure, est une matrice découpée en blocs dans laquelle tous les coefficients au-dessus d'un bloc, respectivement au dessous, sont égaux à zéro. Chaque bloc peut alors être traité indépendamment des autres à l'exception de ceux qui le précèdent. La Figure 44 représente une matrice dans une forme triangulaire supérieure par blocs. Dans le cas de la résolution d'un système d'équations, les équations correspondant au bloc A_1 peuvent être résolues indépendamment du bloc A_2 . La résolution des équations correspondant au bloc A_2 nécessite les résultats des équations du bloc A_1 en utilisant les coefficients du bloc A_{21} .

Pour qu'une matrice puisse être mise dans une forme triangulaire par blocs il faut qu'elle ne soit pas irréductible. Une matrice carrée A d'ordre n est réductible si on a une matrice de permutation P ⁶ telle que :

$$PAP^t = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

A l'inverse, une matrice est dite irréductible s'il n'existe pas une telle permutation. Le théorème suivant nous fournit une méthode simple pour déterminer si nous pouvons obtenir une forme triangulaire par blocs pour notre matrice des coefficients⁷.

Théorème. *une matrice n'est pas réductible si son graphe orienté est fortement connecté.*

5. Nous ne pouvons pas utiliser de méthode tirant parti de la multiplication de matrices creuses car les puissances de C deviennent denses après quelques itérations seulement [100].

6. Une matrice de permutation n'est composée que de vecteurs colonnes dont tous les éléments sont égaux à zéro à l'exception d'un seul élément.

7. Ce théorème et sa preuve sont présentés dans [12].

Le calcul des flux cumulés, ou de l'inverse de la matrice des coefficients, peut donc se réaliser indépendamment pour chaque bloc. Étant donné que les algorithmes de résolution de systèmes d'équations ou de calculs de l'inverse d'une matrice sont relativement coûteux, cette technique nous permet de réduire la complexité générale en effectuant les opérations coûteuses sur des sous-ensembles de la matrice des coefficients.

Dans le cas d'une base de données d'inventaires, l'intérêt de cette décomposition du graphe détaillé, outre l'optimisation du calcul des flux, réside dans l'analyse de la non-calculabilité d'une modélisation. Comme nous l'avons expliqué précédemment, si le rayon spectral de la matrice des coefficients est supérieur à 1, la modélisation n'a physiquement pas de sens et la matrice ne peut-être inversée. Mais cette information n'est donnée que pour tout le graphe détaillé. L'ensemble des valeurs propres d'une matrice est égal à l'union des valeurs propres de ses blocs une fois passée sous la forme d'une matrice triangulaire par bloc [14]. Donc, si l'un des sous-graphe fortement connexe du graphe détaillé ne respecte pas la condition portant sur le rayon spectral, les flux cumulés ne sont pas calculables. Dans le cas où l'inverse d'un bloc n'est pas calculable, nous pouvons quand même calculer l'inverse des blocs amonts. De plus, nous pouvons détecter plus finement quelle partie d'une modélisation rend l'ensemble non-calculable. Notons que cette proposition n'est valable que si le graphe détaillé ne comporte pas qu'une seule composante fortement connexe.

Pour déterminer la forme triangulaire par bloc de la matrice des coefficients, nous pouvons utiliser l'algorithme de Dulmage-Mendelshon [24, 72]. Comme nous n'avons à décomposer qu'une matrice carrée, seule la dernière étape basée sur la décomposition en composantes fortement connexes du graphe détaillé est réellement utile [23]. La méthode de Tarjan [85] se base sur un parcours en profondeur du graphe et permet pour une complexité en $O(n + m)$ avec n nœuds et m arcs, d'obtenir les composantes fortement connexes ordonnées topologiquement. Le graphe réduit du graphe détaillé $G = (V, E)$ est noté $G_{SCC} = (V_{SCC}, E_{SCC})$ où V_{SCC} contient un nœud pour chaque composante fortement connexe de G . L'ensemble E_{SCC} contient un arc entre deux nœuds u et v s'il existe un arc entre deux nœuds des composantes fortement connexes correspondantes. L'algorithme suivant donne les grandes étapes de ce traitement :

Algorithme 6: Décomposition du graphe détaillé G

Entrées : G

Construire G_{SCC} à partir de G

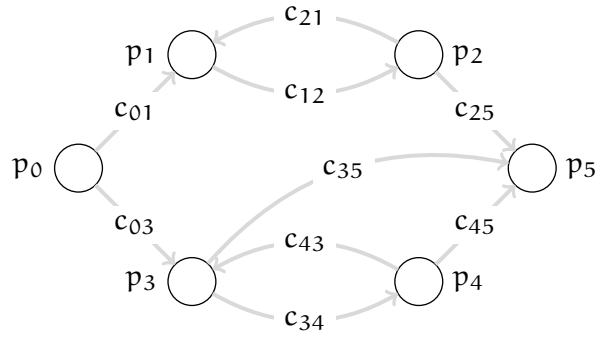
Trier topologiquement G_{SCC}

pour chaque $v \in V_{SCC}$ **faire**

$C_u \leftarrow$ matrice des coefficients de v

si $\rho(C_u) < 1$ **alors**

 Calculer C_u^{-1}

FIGURE 45: Graphe détaillé entre six processus p_i avec $0 \leq i \leq 5$.

6.2.2 Exemple de décomposition

Soit G un graphe détaillé de processus tel que présenté dans la Figure 45. La matrice des $(I - C)$ que nous extrayons de ce graphe est :

$$(I - C) = \begin{pmatrix} 1 & c_{01} & 0 & c_{03} & 0 & 0 \\ 0 & 1 & c_{12} & 0 & 0 & 0 \\ 0 & c_{21} & 1 & 0 & 0 & c_{25} \\ 0 & 0 & 0 & 1 & c_{34} & c_{35} \\ 0 & 0 & 0 & c_{43} & 1 & c_{45} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Une fois appliqué, un algorithme de décomposition en composantes fortement connexes et un algorithme de tri topologique sur ce graphe, nous obtenons le graphe G_{SCC} de la Figure 46. Les flux cumulés de chaque composante fortement connexe peuvent ensuite être calculés dans l'ordre topologique de ce graphe en utilisant la matrice des coefficients correspondant à ces composantes : V_{SCC_0} puis V_{SCC_1} , etc.

6.2.3 Calcul des flux cumulés dans un système dynamique

Nous souhaitons recalculer les flux cumulés dans une base d'inventaires suite à chaque modification apportée sur le modèle des relations de dépendance. La décomposition en composantes fortement connexes peut être utile pour optimiser les calculs suite à chaque modification du graphe détaillé. Après chaque modification, nous pouvons déterminer s'il est nécessaire de recalculer l'ensemble des flux cumulés ou seulement ceux d'un sous-ensemble du graphe détaillé.

Si nous calculons l'inverse des matrices des composantes fortement connexes du graphe détaillé au lieu de directement calculer les flux cumulés, nous pouvons recalculer plus rapidement les flux cumulés suite à une modification sur l'inventaire. Selon le type de modifications apportées sur le graphe détaillé, nous pouvons être amenés à recalculer certains de ces inverses.

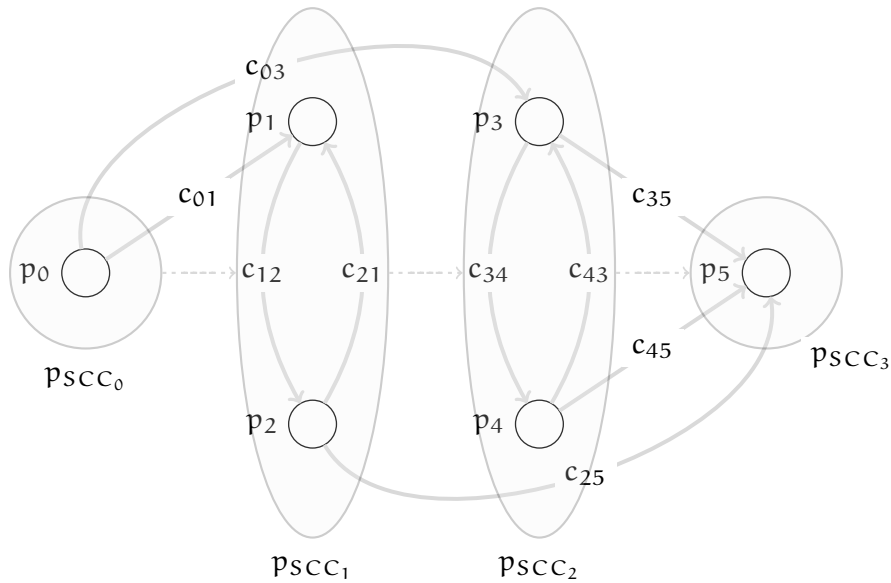


FIGURE 46: Décomposition en composantes fortement connexes du graphe détaillé de la Figure 45. Les nœuds p_{SCC_i} correspondent aux nœuds du graphe G_{SCC} construit à partir de G . Les arcs en pointillés sont les arcs de G_{SCC} . Les nœuds du graphe G_{SCC} sont dans l'ordre topologique.

Nous distinguons six types de modifications qui nécessitent de recalculer tout ou partie du graphe détaillé. Nous détaillons les actions à réaliser pour chaque type de modification dans la section suivante.

6.2.3.1 Prise en compte des différents types de modifications

MODIFICATION DU FLUX ÉLÉMENTAIRE D'UN PROCESSUS Il n'est pas nécessaire de réaliser d'inversion de matrice. Nous nous contentons de multiplier l'inverse d'une matrice d'une composante fortement connexe par le vecteur des flux élémentaires des processus qui la composent. Il faut réaliser cette opération pour la composante à laquelle appartient le processus modifié ainsi qu'aux composantes en aval.

MODIFICATION DE LA VALEUR D'UN COEFFICIENT Si le coefficient relie des nœuds appartenant à deux composantes différentes, il faut recalculer les flux cumulés en aval mais il n'est pas nécessaire de réaliser des inversions de matrices. Si le coefficient relie deux nœuds d'une même composante, il faut recalculer l'inverse de la matrice correspondant à cette composante. Puis recalculer les flux cumulés des processus de cette composante et ceux des composantes aval.

AJOUT D'UN COEFFICIENT Si le nouveau coefficient relie deux nœuds qui appartiennent à la même composante, il suffit de recalculer l'inverse de la matrice des coefficients de cette composante. Dans le cas où les deux nœuds appartiennent à deux composantes différentes, nous distinguons deux cas :

- s’il n’y a pas d’arc dans le graphe détaillé dans le sens inverse de ce nouvel arc, il suffit de recalculer les flux cumulés des composantes en aval de cet arc ;
- s’il y a un arc dans le graphe détaillé dans le sens inverse de ce nouvel arc, les deux composantes doivent être fusionnées et il faut calculer l’inverse de la matrice des coefficients de cette nouvelle composante.

SUPPRESSION D’UN COEFFICIENT Si le coefficient relie deux composantes fortement connexes, nous n’avons qu’à recalculer les flux cumulés des composantes en aval. Mais, si le coefficient relie deux nœuds de la même composante, nous devons déterminer si la composante peut être décomposée en plusieurs composantes.

AJOUT D’UN PROCESSUS Un nouveau processus ne sera relié à aucun autre processus et fera donc partie d’une nouvelle composante fortement connexe. Il n’est donc pas nécessaire de recalculer des flux cumulés.

SUPPRESSION D’UN PROCESSUS Le traitement de la suppression d’un processus peut se rapporter à la suppression de tous ses arcs entrants et sortants.

6.2.3.2 Discussion

L’intérêt de cette proposition est cependant limité par deux aspects liés au degré d’inter-connexions des activités humaines et à l’efficacité des méthodes de calcul existantes.

Nous ne sommes pas assurés d’obtenir un graphe détaillé comportant plus d’une composante fortement connexe ou un nombre suffisant de grandes composantes pour justifier la décomposition. En d’autres termes, il est possible que toutes les activités humaines soient inter-connectées. Pour confirmer ou infirmer ce point, nous avons étudié la topologie des graphes issue de trois tableaux d’Entrée/Sortie (E/S) économiques. Ces tableaux sont réalisés par différents bureaux de statistiques économiques et contiennent les interactions entre tous les secteurs économiques d’un territoire donné. Nous avons étudié le tableau d’E/S de l’Organisation de Coopération et de Développement Économique (OCDE) qui prend en compte tous les pays de l’OCDE⁸, le tableau d’E/S du Bureau of Economic Analysis qui se limite à l’économie américaine⁹ et le tableau d’E/S de l’agence d’analyse économique anglais qui s’applique au Royaume-Uni¹⁰.

Les résultats de cette analyse sont présentés dans le tableau 3. A ce niveau de décomposition des activités économiques, nous pouvons en conclure que nous n’avons pas suffisamment de composantes fortement connexes pour justifier l’intérêt de notre proposition. Mais la décomposition de ces tableaux d’E/S est bien moins précise que celle d’une base d’inventaires¹¹. Nous ne pouvons donc

8. Accessible sur <http://stats.oecd.org>

9. Accessible sur <http://www.bea.gov/itable>

10. Accessible sur <http://www.statistics.gov.uk>

11. Il y a 194 activités différentes dans le tableau d’E/S de l’OCDE et plusieurs milliers dans la base Ecoinvent.

	OCDE	BEA	UK
<i>Nombre de noeuds</i>	194	65	113
<i>Nombre de composantes fortement connexes</i>	33	4	11
<i>Taille de la composante la plus grande</i>	162	62	103

TABLE 3: Comparaison des graphes des tableaux d'Entrée/Sortie économiques de différentes agences d'analyse économique.

pas confirmer que les activités humaines sont toutes inter-connectées pour une décomposition plus précise.

L'autre point limitant l'intérêt de notre approche est lié à l'efficacité des méthodes de calcul déjà existantes. Nous avons réalisé des tests avec des algorithmes de résolution itérative de systèmes d'équations disponibles dans la librairie BLAS [3] sur des matrices issues de la Sparse Matrix Collection de l'Université de Floride [22]. Pour des matrices comportant environ 400 000 nœuds et avec une densité de moins de 1%, nous obtenons des temps de calcul inférieurs à 100 ms sous Linux sur un PC équipé d'un Intel Xéon E31245 à 3.30GHz avec 8Go de mémoire vive. My C-sense ne réalisant pas de simulation de Monte-Carlo pour calculer les flux cumulés dans une base de facteurs d'émissions de gaz à effet de serre, une seule itération suffit suite à une modification unitaire sur le modèle. Nous n'avons donc pas jugé utile de réaliser, pour l'instant, des expérimentations sur l'efficacité de notre proposition pour l'optimisation des calculs.

6.3 SYNTHÈSE

Nous avons vu que la méthode la plus efficace pour calculer les flux cumulés d'un inventaire est d'utiliser un algorithme de résolution itératif. Le découpage en composantes fortement connexes du graphe détaillé permet d'obtenir une information plus précise sur la raison de non-calculabilité des flux cumulés. Cette méthode permet aussi d'optimiser le calcul des flux cumulés dans un système dynamique. Mais l'intérêt de cette proposition est limité par le nombre de composantes fortement connexes, *i.e.* par le degré d'inter-connexion des activités humaines, dans le graphe détaillé d'une base d'inventaires.

PROTOTYPE(S)

Dans le cadre de cette thèse, nous avons développé deux prototypes pour valider certains choix de notre méthode de modélisation sémantique multi-niveaux d'une base d'inventaires. La première version a été développée pour tester certains points de la proposition initiale : le regroupement des processus et des coefficients et l'indexation par des mots-clefs. A l'origine, notre proposition consistait à indexer les éléments de la base par des termes contenus dans une taxonomie et on ne pouvait exprimer que des macro-relations normales. Les dimensions étaient alors exprimées comme les intersections des fils d'un ensemble de termes de la taxonomie. Mais l'utilisation d'une taxonomie ne permettait pas d'exprimer clairement les relations entre les concepts servant à indexer les éléments de la base. Nous avons donc décidé de construire une ressource sémantique plus proche de la notion d'ontologie au sens de D. L. McGuiness [53]. Le deuxième prototype reprend tous les points de notre proposition telle que nous l'avons présentée dans ce document. My C-sense a ensuite développé une application opérationnelle de cette base, avec pour objectif, à court terme, de servir de référentiel des données de l'ADEME pour son application de comptabilité carbone. Étant donné la faible transparence sur la composition des facteurs d'émissions de l'ADEME, elle n'implémente pas la modélisation multi-niveaux. Pour l'instant, seul l'indexation sémantique et la notion de groupes ont été repris dans l'application opérationnelle. Ces deux notions ont en revanche été étendues pour mieux exprimer la sémantiques des éléments de la base. La deuxième version de notre prototype a été présenté à CIKM 2012 [7].

Les deux prototypes sont des applications web développées en PHP construites sur Zend Framework côté serveur. La partie client utilise les bibliothèques Javascript Yahoo UI, jQuery et Protovis [11]. L'application opérationnelle utilise les mêmes technologies que les prototypes et fournit une API pour s'interfacer avec l'application de comptabilité carbone.

Nous détaillons dans ce chapitre les interfaces et les fonctionnalités du deuxième prototype. Puis nous présentons les évolutions introduites dans l'application opérationnelle et ses interactions avec l'application de comptabilité carbone pour mettre en avant l'intérêt opérationnel de l'indexation sémantique.

7.1 INTERFACES

7.1.1 *Exploration de la base d'inventaires*

Le prototype est composé d'une interface de consultation et d'une interface d'édition de la base. L'interface de consultation contient un moteur de



FIGURE 47: Écran d'accueil de l'interface de consultation. Cet écran contient des informations sur le contenu de la base d'inventaires courante, un moteur de recherche et un accès aux différentes bases d'inventaires de l'application.

recherche permettant de trouver des groupes et des éléments individuels (processus et coefficients) à partir des mots-clés les indexant. La Figure 47 contient une capture d'écran de l'accueil de l'interface de consultation.

Cette interface contient aussi un module de visualisation d'un élément (groupe ou élément individuel) découpé en deux panneaux, la Figure 48 présente une capture d'écran de l'interface de consultation mettant en avant les différentes zones du module de visualisation.

Le premier panneau (Panneau détails de la figure) présente des informations sur l'élément en cours de consultation : émissions de gaz à effet de serre, indexation et unité de mesure pour les processus ; valeur numérique, indexation et unité de mesure pour les coefficients. Les groupes sont présentés sous la forme d'un tableau multi-dimensionnel affichant l'impact carbone des processus ou les valeurs numériques des coefficients dans chaque cellule, la Figure 49b contient une capture d'écran de la visualisation d'un groupe de processus.

Le deuxième panneau affiche le graphe détaillé ou le macro-graphe et permet de naviguer au sein de ces graphes. L'affichage de ces graphes reprend la méthode de visualisation radiale d'un graphe [43] : l'affichage est centré sur le processus ou le groupe de processus consulté, les nœuds amonts et avals sont respectivement affichés sur un demi-cercle à gauche et à droite du nœud central. La Figure 49a illustre cet affichage. Les nœuds plus distants sont affichés en utilisant la même méthode mais en ne conservant que les processus amonts pour les nœuds en amont et les processus avals pour les nœuds en aval.

La navigation se fait en cliquant sur un nœud du graphe qui devient alors l'élément central, le contenu du panneau d'information est mis à jour avec les

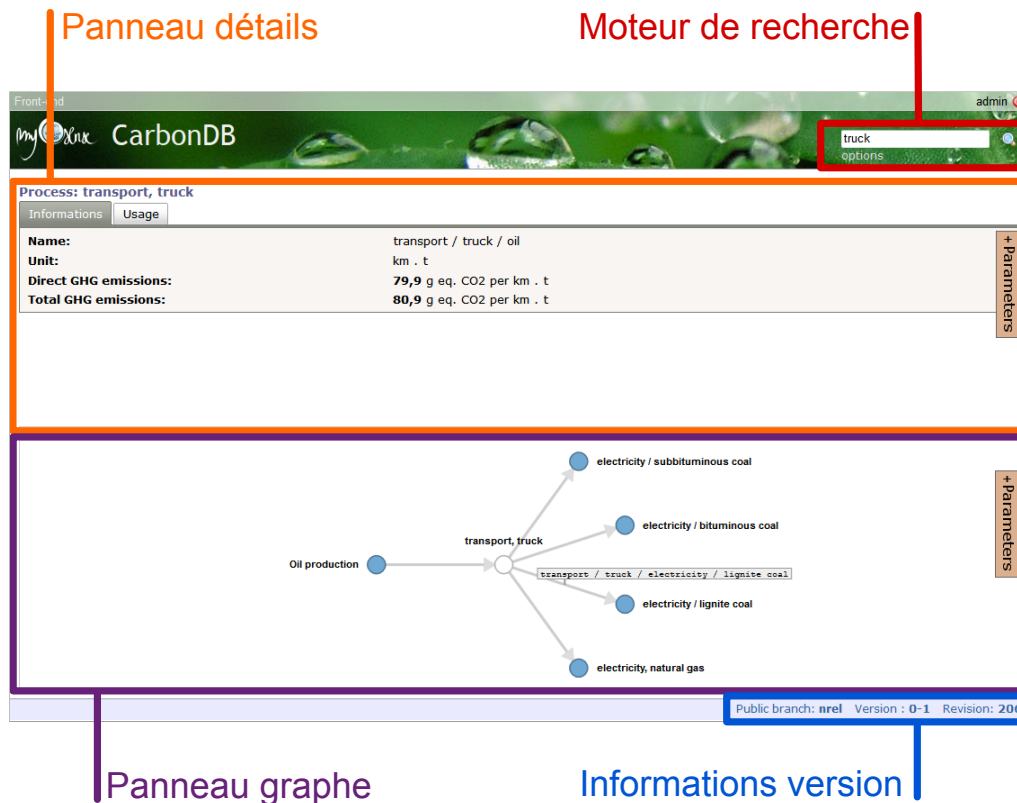


FIGURE 48: Interface d'exploration de la base d'inventaires.

données du nœud central. Si le nœud central affiché dans le macro-graphe est un processus individuel, l'utilisateur peut passer à l'affichage du graphe détaillé centré sur ce nœud. Si le nœud central est un groupe de processus, l'utilisateur peut sélectionner un processus affiché dans le tableau multi-dimensionnel du panneau d'informations pour passer au graphe détaillé. Quand l'affichage est centré sur un processus, le panneau d'informations contient un onglet affichant la liste des groupes le référant ce qui permet de passer au macro-graphe. Les coefficients et les groupes de coefficients sont affichés au survol sur les arcs des graphes. Si l'utilisateur sélectionne l'un de ces arcs, le panneau d'information est mis à jour avec les informations du coefficient ou du groupe de coefficients.

7.1.2 Édition de la base d'inventaires

L'interface d'édition contient plusieurs outils d'édition des différents éléments de la base :

- un éditeur pour l'ontologie permettant d'ajouter des mots-clés, des prédicats et de créer des relations entre des mots-clés (la Figure 50 contient une capture d'écran de cet éditeur) ;
- un module d'édition des groupes basé sur l'expression des dimensions des groupes ;

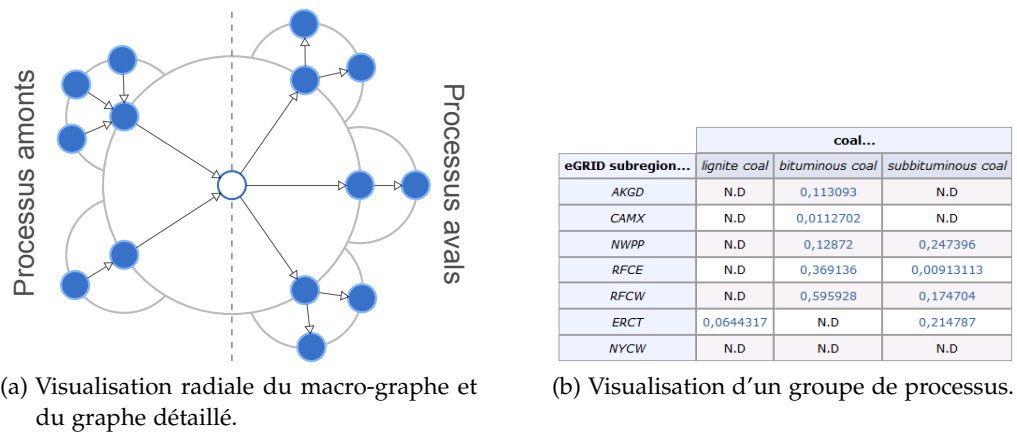
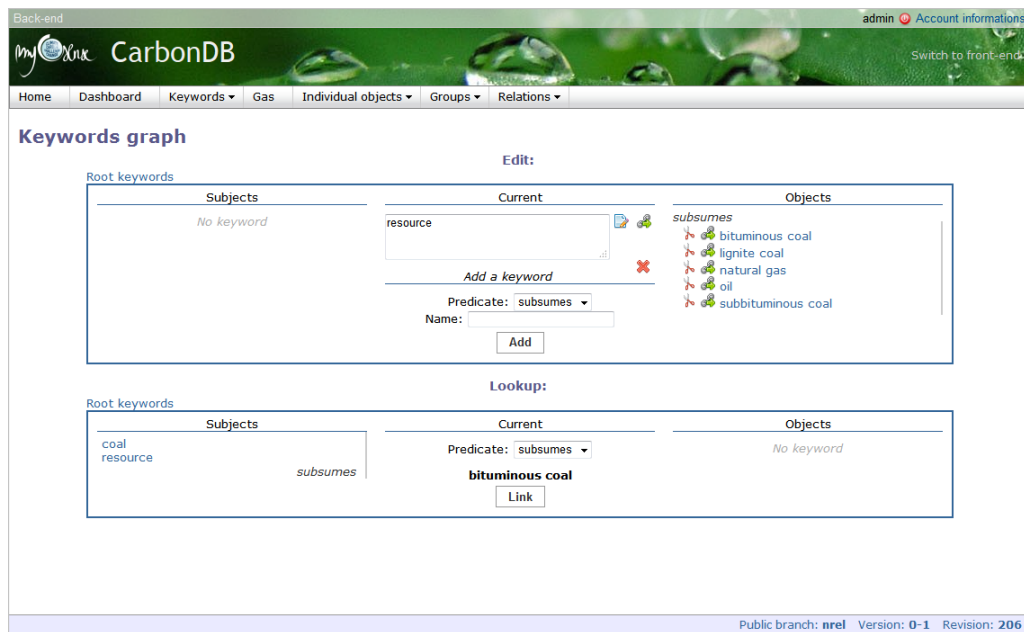


FIGURE 49: Éléments de l'interface d'exploration de la base d'inventaires.



- un module d'édition des valeurs des coefficients et des flux élémentaires des processus (la Figure 51 contient une capture d'écran de l'édition des flux élémentaires d'un processus) ;
- un module d'édition des relations entre des groupes de processus ;
- un tableau de bord contenant des informations statistiques sur le contenu de la base (nombre de processus, nombre de groupes, etc.).

Dans le prototype, la conversion du macro-graphe se fait en temps différé en trois étapes : suppression des relations détaillées, conversion des macro-relations, calcul des flux cumulés des processus. Notons que toutes les relations détaillées sont issues de la traduction des macro-relations, il n'est pas possible d'éditer directement une relation détaillée. Nous avons fait la distinction entre les relations qui sont exprimées par l'éditeur de la base, que nous nommons *relations sources*, et les relations qui sont issues de la conversion de ces relations

The screenshot shows the CarbonDB web application interface. At the top, there is a navigation menu with options like Home, Dashboard, Keywords, Gas, Individual objects, Groups, and Relations. The main content area is titled 'Processus' and features a 'Filter' section on the left. The central part of the interface is a form for editing a process. It includes a 'Name' field containing 'transport, truck', a 'Keywords' list with 'transport', 'truck', and 'oil', and a 'Unit' field set to 'km.t'. Below this is an 'Impacts' table with the following data:

ID	Gas	Unit	Mass	Uncertainty (%)	Flow origin	Data origin
331	undistinguished	g	80.876	0		
330	undistinguished	g	1	0		indirecte
329	undistinguished	g	79.876	0		directe
328	undistinguished	g	1	0		indirecte
327	undistinguished	kg	0.079876	0	source	directe

The footer of the application indicates 'Public branch: nrel Version: 0-1 Revision: 206'.

FIGURE 51: Édition des flux élémentaires d'un processus.

sources, que nous nommons *relations dérivées*. Les relations sources peuvent être exprimées entre des groupes de processus ou des processus individuels alors que les relations dérivées correspondent uniquement à des relations détaillées. Le macro-graphe peut donc contenir des processus individuels qui, dans le formalisme que nous avons exposé précédemment, se comportent comme des groupes à un élément.

7.2 FONCTIONNALITÉS TRANSVERSES

Nous avons intégré plusieurs fonctionnalités transverses dans le prototype :

- historisation des actions utilisateurs ;
- gestion de plusieurs bases d'inventaires ;
- gestion des droits d'accès des utilisateurs.

Les fonctionnalités d'historique et de gestion multi-bases permettent à des utilisateurs différents d'éditer chacun leurs propres bases ou de collaborer sur des bases communes. L'historique des actions utilisateurs permet de consulter une base dans un état antérieur¹. La Figure 52 contient une capture d'écran de l'interface listant les actions utilisateurs et qui permet de consulter la base dans un état antérieur. Une nouvelle base peut être vide ou être une copie d'une base existante (dans sa version courante ou dans une version antérieure) et se voit attribuer une visibilité reprenant la terminologie de la programmation objet : publique, privée ou protégée. Les utilisateurs ont des droits d'accès différents sur chaque base : éditeur, utilisateur ou propriétaire. Une base publique sera visible par tout les utilisateurs (y compris les utilisateurs anonymes), une base

1. Nous avons élaboré l'historique de façon à pouvoir rejouer toutes les actions utilisateurs et nous conservons l'état initial de chaque base.

Back-end admin Account informations
CarbonDB Switch to front-end

Home Dashboard Keywords Gas Individual objects Groups Relations

Révisions

Filter :

<< first < previous Page : 1 / 11 next > end >> | Per page : 20

ID	Class	Method	Date	User	Revert
206	GrapheProcessus	calculGraphe	2012-10-26 16:09:30	Bertin Benjamin	
205	RelationSource	ajouter	2012-10-26 16:09:24	Bertin Benjamin	
204	GrapheProcessus	calculGraphe	2012-10-26 16:06:42	Bertin Benjamin	
203	Processus	modifier	2012-10-26 16:06:38	Bertin Benjamin	
202	Flux	modifier	2012-10-26 16:06:36	Bertin Benjamin	
201	GrapheProcessus	calculGraphe	2012-10-26 15:31:12	Bertin Benjamin	
200	Processus	renommer	2012-10-26 15:31:09	Bertin Benjamin	
199	Flux	modifier	2012-10-26 15:31:07	Bertin Benjamin	
198	GrapheProcessus	calculGraphe	2012-10-26 15:28:36	Bertin Benjamin	
197	Processus	renommer	2012-10-26 15:28:27	Bertin Benjamin	
196	Flux	modifier	2012-10-26 15:28:25	Bertin Benjamin	
195	Flux	modifier	2012-10-26 15:28:22	Bertin Benjamin	
194	GrapheProcessus	calculGraphe	2012-10-26 15:25:02	Bertin Benjamin	
193	Coefficient	supprimer	2012-10-26 15:18:38	Bertin Benjamin	
192	Coefficient	supprimer	2012-10-26 15:18:24	Bertin Benjamin	
191	Coefficient	supprimer	2012-10-26 15:18:19	Bertin Benjamin	

Public branch: nrel Version: 0-1 Revision: 206

FIGURE 52: Historique des actions utilisateurs pour une base d'inventaire.

protégée uniquement par les utilisateurs enregistrés et une base privée uniquement par les utilisateurs ayant les bons droits d'accès.

7.3 APPLICATION OPÉRATIONNELLE

L'application qui a été développée à la suite de ce prototype reprend certains points de notre approche et étend certaines notions. L'indexation sémantique a été étendue pour reprendre la notion d'étiquetage sémantique. L'indexation d'un élément utilise un mot-clef supplémentaire pour préciser le sens donné par le mot-clef d'indexation pour éviter certaines confusions. Nous avons aussi levé la contrainte de l'unicité d'un élément correspondant à un ensemble de mots-clefs, les éléments référencés dans un groupe sont alors choisis par l'éditeur s'il y a plusieurs éléments éligibles. Cette modification apportée au modèle n'a pas d'impact sur le traitement des macro-relations car il n'y a qu'un seul élément choisi par cellule d'un groupe.

L'application de comptabilité carbone de My C-sense récupère les facteurs d'émissions et les coefficients *via* une API. Elle offre des méthodes de récupération des éléments en fonction de leur indexation et une méthode de récupération des mots-clefs d'une dimension d'un groupe. Les dimensions servent à élaborer des formulaires de collecte de données comptables automatiquement. Par exemple, nous disposons d'un groupe des processus de chauffage d'un bâtiment en fonction de son mode de chauffage (gaz, électrique, etc.). Un formulaire de saisie possède un champ de sélection permettant de choisir un mot-clef de la dimension des modes de chauffage. Ce choix sert ensuite à sélectionner un processus parmi tous ceux présent dans le groupe pour calculer l'impact du chauffage d'un bâtiment en fonction de sa surface. Finalement, lors de la phase d'analyse des résultats et si nous avons plusieurs bâtiments, les mots-

clefs des processus qui ont été utilisés servent à alimenter une dimension d'un cube d'analyse.

7.4 SYNTHÈSE

Le premier prototype que nous avons développé a permis d'affiner notre approche. Le deuxième prototype a donné lieu à un développement plus important de la partie consultation des données et a abouti à la conception d'une interface de visualisation et de navigation dans le macro-graphe et le graphe détaillé. Le développement d'une version opérationnelle par My C-sense a validé l'intérêt de l'utilisation de la sémantique pour l'indexation des éléments de la base. Mais l'utilisation de la modélisation sémantique multi-niveaux dans un contexte opérationnel nécessitera d'intégrer des données issues de bases plus transparentes que le guide des facteurs d'émissions de l'ADEME.

La modélisation des processus métier, ou Business Process Modeling, est relativement proche de la modélisation des relations de dépendance entre les processus de l'ACV. On retrouve dans cette discipline une problématique liée à la compréhensibilité des modèles de processus métier ainsi que des approches pour leurs simplifications qui se rapprochent de notre proposition sur la modélisation multi-niveaux d'un inventaire en cycle de vie. Nous présentons dans ce chapitre une synthèse bibliographique des travaux connexes portant sur la compréhensibilité et l'abstraction de modèles de processus métier. Dans la première section, nous introduisons la gestion des processus métier et les méthodes de modélisation. Dans la deuxième section, nous présentons des travaux portant sur la compréhensibilité des modèles de processus métier. Dans la troisième section, nous détaillons des travaux sur des méthodes d'abstraction de processus métier permettant de simplifier leurs modèles.

8.1 GESTION DES PROCESSUS MÉTIER

La gestion des processus métier, ou Business Process Management (BPM), a pour objectif de mettre à profit les systèmes d'information pour la réalisation et l'amélioration des processus métier d'une entreprise. D'après [18] et [39], un processus métier, ou Business Process (BP), se définit comme un ensemble d'activités logiquement coordonnées pour fournir un résultat à un client ou au marché. On retrouve donc un objectif similaire à l'un des aspects de l'analyse en cycle de vie : modéliser les interactions entre les activités humaines. Mais les finalités de ces deux approches sont complètement différentes. Dans l'analyse en cycle de vie, la finalité est de calculer les impacts environnementaux alors que celle du BPM est de mettre en œuvre ces activités. En d'autres termes, l'ACV décrit l'interaction entre des processus pour calculer leurs effets sur l'environnement alors que le BPM décrit le comportement des processus pour leur réalisation.

Le cycle de vie du BPM se décompose en quatre phases : modélisation, configuration, réalisation et diagnostique [93, 50] (voir Figure 53). La phase de modélisation d'un processus métier correspond à la modélisation des flux économiques de l'analyse en cycle de vie. Mais comme cette modélisation sert, in fine, à exécuter un processus métier, sa formalisation est beaucoup plus riche et complexe que la simple expression de relations de dépendance entre des activités humaines.

La modélisation des processus métier peut se décomposer en trois niveaux : le premier correspond au méta-modèle, c'est à dire au langage de modélisation ; le deuxième au modèle d'un processus métier et le dernier au différentes ins-

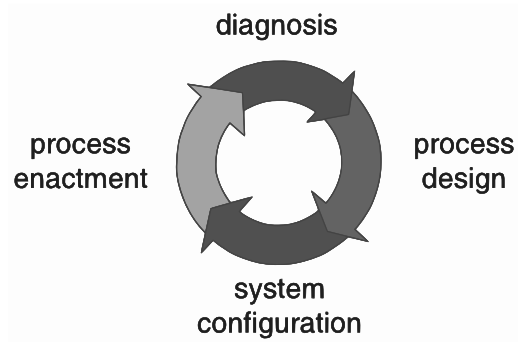


FIGURE 53: Cycle de vie du BPM repris de [50].

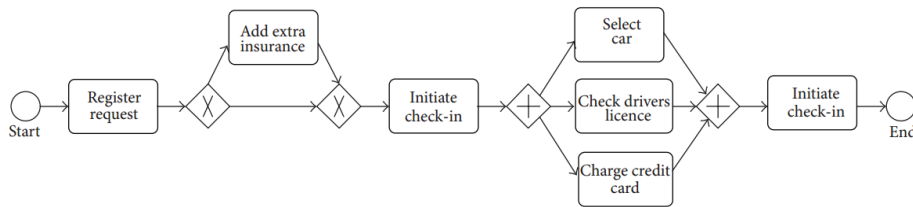
tances de ce modèle, c'est-à-dire à ses différentes exécutions. D'après [92], on distingue trois classes de langages de modélisation d'un processus métier qui servent plusieurs objectifs :

- les *langages formels* permettent de vérifier la justesse de la modélisation. La modélisation formelle la plus connue et la plus étudiée utilise les réseaux de Petri [68, 91];
- les *langages conceptuels* sont des langages de plus haut niveau et permettent aux experts métier de s'affranchir de la sémantique rigoureuse des langages formels. Les langages les plus utilisés sont EPC (Event-Driven Process Chain) [47] et, plus récemment, BPMN (*Business Process Modeling Notation*) [66];
- les *langages d'exécution* servent à décrire les éléments techniques des processus métier en vue de leur exécution, tels que BPEL (*Business Process Execution Language*) [4].

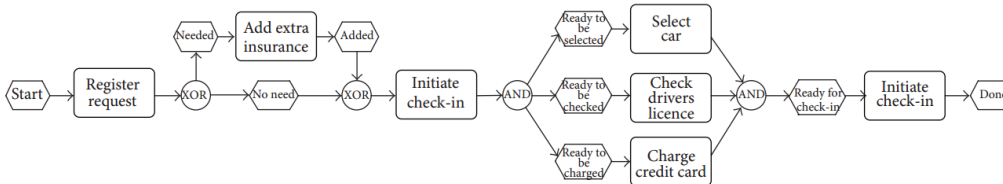
Les langages conceptuels, notamment EPC et BPMN, sont les plus utilisés pour modéliser des processus métier [60]. Ils reposent sur des notions communes :

- un processus métier contient une suite ordonnée d'activités atomiques (nommée tâches dans la terminologie BPMN et fonctions dans la terminologie EPC);
- des connecteurs logiques peuvent être utilisés pour relier les différentes activités atomiques;
- un processus métier a un début et une fin.

L'équivalent d'un processus de l'analyse en cycle de vie correspond à une activité atomique. Mais, contrairement à l'ACV, la modélisation d'un processus métier n'a pas pour objectif de modéliser toutes les relations de dépendance entre les différentes activités humaines. La Figure 54 présente un exemple de processus métier modélisé avec BPMN (54a) et EPC (54b). Les connecteurs logiques permettent de définir si un ensemble de tâches doivent être exécutées en parallèle (connecteur AND) ou si seulement certaines d'entre elles doivent être exécutées (connecteur OR et XOR). Les tâches qui sont liées entre elles sans connecteur logique sont exécutées séquentiellement.



(a) Modèle BPMN.



(b) Modèle EPC.

FIGURE 54: Exemples de modèles de processus métier avec deux langages de modélisation conceptuels (BPMN et EPC) repris de [92].

Presque tous les langages de modélisation permettent de modéliser les processus métier de façon modulaire et donc de définir des niveaux de détail différents (ou permettre la réutilisation de modèle). Par exemple, dans BPMN une activité n'est pas forcément atomique et peut être un sous-processus.

Certains travaux [95, 2] mettent à profit cette modularité et proposent des méthodes basées sur plusieurs niveaux d'abstraction de la tâche jusqu'à une carte des processus de l'entreprise. Les niveaux d'abstraction les plus élevés ne reprennent pas les mêmes langages de modélisation que les niveaux détaillés mais servent essentiellement à mettre en avant les relations entre chaque processus. Cette approche permet de simplifier la compréhension pour chaque acteur impliqué en ne présentant que l'information dont il a besoin. Mais, comme avec l'utilisation de sous-processus, il n'y a pas de transformation automatique des niveaux supérieurs vers les niveaux inférieurs. De plus, contrairement à notre proposition, les niveaux d'abstraction élevés agrègent généralement des séquences d'activités ou des processus entiers.

Ghose et al. ont proposé d'intégrer des données d'impacts carbone dans des modèles de processus métier (sur une modélisation BPMN) [44, 32]. Ces travaux et ceux qui ont suivi [63, 62] ont pour objectif de permettre de réduire l'impact carbone des processus au cours de la phase de modélisation. Nowak et al. ont aussi proposé une méthode de visualisation des impacts de chaque sous-processus et activités dans un modèle de processus. Mais ces propositions se limitent à la modélisation des impacts directs des processus et ne sont utiles que pour la modélisation d'un processus métier et elles ne visent pas à modéliser les interactions entre toutes les activités humaines.

8.2 COMPRÉHENSIBILITÉ DES MODÈLES DE PROCESSUS MÉTIER

Plusieurs études récentes se sont intéressées à la question de la compréhensibilité et, par extension, à la question de la maintenabilité des modèles de processus métier [57, 31]. Nous présentons ici ces travaux et leurs principales conclusions qui rejoignent la problématique de la compréhensibilité des inventaires en cycle de vie.

Mendling et al. ont étudié les facteurs qui impactent la compréhensibilité d'un modèle de processus métier [57] à l'aide d'un questionnaire auprès d'un panel d'étudiants et d'une interview d'experts en modélisation de processus métier. Leur étude a consisté à présenter plusieurs processus métier à des étudiants, modélisés avec EPC¹, puis à leur demander de répondre à un questionnaire comportant des questions relatives à la compréhension de l'ordonnement des activités. Ils ont utilisé quatre modèles de processus métier avec chacun trois variantes obtenues en remplaçant un connecteur logique par un autre (par exemple, un AND par un OR ou un XOR). Ils ont cherché à déterminer si certains facteurs ont une incidence sur la compréhensibilité telle que le niveau d'expertise des participants, le nombre de liaisons entre les éléments du processus métier ou encore le nombre de connecteurs logiques. Certains facteurs ont été écartés tels que la structuration du modèle, sa séquentialité ou sa taille (*i.e.* : son nombre d'activités et de jonctions). C'est pourquoi tous les processus métier qu'ils ont utilisés comportent le même nombre de nœuds, n'ont aucune structure et ne sont pas modulaires. Les résultats de leur étude montrent que *le nombre d'arcs entre les éléments d'un modèle de processus métier est inversement proportionnel à sa compréhensibilité*. Il en va de même pour le degré (entrant et sortant) des connecteurs. Les auteurs ont en revanche établi que le niveau d'expertise n'a que peu d'influence sur la compréhensibilité d'un modèle.

Dans [74], Reijers et al. ont réalisé une étude sur l'intérêt de la modularité dans la modélisation des processus métier. Leur étude a été réalisée auprès d'experts en modélisation de processus métier et portait sur la compréhensibilité de plusieurs modèles dans deux versions : une "à plat" et une modulaire. Les résultats de leur étude ont mis en évidence l'intérêt de la modularité pour la compréhensibilité des modèles de processus métier.

Ces mêmes auteurs ont proposé sept recommandations pour la modélisation de processus métier [59]. Deux de ces recommandations sont liées à la taille du modèle : *R1* : utiliser le moins d'éléments possibles et *R7* : décomposer le modèle s'il a plus de 50 éléments. Pour un modèle de processus métier comportant plus de 50 éléments, le taux d'erreurs tend à être plus grand que 50% [77, 56]. Les autres recommandations concernent le nombre de chemins possibles, la présence d'un point d'entrée et d'un point de sortie, la structuration du modèle, la réduction du nombre de connecteurs OR et l'utilisation de verbes

1. Le choix d'EPC a été fait conformément aux résultats de l'étude présentés dans [77] qui recommande l'utilisation d'EPC pour faciliter la compréhension d'un modèle de processus métier.

pour décrire les activités.

Dans [35] et [31] des métriques pour déterminer la complexité des modèles de processus métier sont proposées. Les auteurs ont appliqué des métriques de complexité existantes en développement logiciel. Parmi celles-ci on retrouve des métriques qui peuvent aussi s'appliquer au domaine de l'analyse en cycle de vie telles que le nombre d'activité d'un modèle (qui est le pendant du nombre de lignes de code) ou le degré de modularité calculé en fonction du *fan-in* et *fan-out*².

8.3 ABSTRACTION DE MODÈLES DE PROCESSUS MÉTIER

L'abstraction de modèles de processus métier a pour objectif de réduire la complexité d'un modèle en créant un ou plusieurs niveaux d'abstraction et donc d'augmenter sa compréhensibilité. Les approches que nous décrivons dans cette section ont comme point commun de créer un modèle abstrait à partir d'un modèle complet ; en d'autres termes ce sont des approches du bas vers le haut. Notre proposition sur la modélisation multi-niveaux apporte aussi une solution pour la création d'un modèle abstrait sur un modèle détaillé, mais dans notre cas nous proposons une approche du haut vers le bas.

Dans [79] et [81], Smirnov et al. ont proposé une définition formelle et une étude de cas sur l'utilisation de l'abstraction de processus métier. Ils définissent l'abstraction de processus métier comme étant une opération sur un modèle qui préserve les propriétés du processus métier qui sont essentielles pour un besoin précis, tout en laissant de côté les aspects non-importants pour ce besoin. Comme nous l'avons vu jusqu'à présent, l'abstraction d'un modèle est réalisable en utilisant les outils de modélisation classiques pour obtenir plusieurs niveaux de détail en utilisant, par exemple, la notion de sous-processus ou la notion de carte de processus. Mais ces méthodes nécessitent de maintenir la cohérence entre tous les niveaux de détail. Les méthodes d'abstraction de modèles de processus métier ont pour objectif de proposer des méthodes automatiques d'abstraction du modèle détaillé d'un processus métier. Les auteurs ont identifié dans la littérature deux méthodes d'abstraction d'un modèle : l'élimination ou l'agrégation d'activités. La Figure 55 illustre ces deux méthodes d'abstraction.

Ils ont identifié 12 cas d'utilisation qu'ils ont proposés à des experts en modélisation de processus métier pour qu'ils déterminent leur intérêt. Les experts ont exprimé un intérêt plus grand pour trois de ces cas d'utilisation. Les deux premiers cas sont liés à l'exécution des processus métier : le premier vise à afficher dans un modèle de processus métier les activités les plus fréquemment

2. Le *fan-in* correspond au nombre de sous-processus utilisés dans un processus métier. Le *fan-out* correspond au nombre d'utilisations d'un processus métier donné dans d'autres processus métier. Une métrique conjuguant ces deux valeurs permet de rendre compte de la complexité structurelle d'un modèle de processus métier.

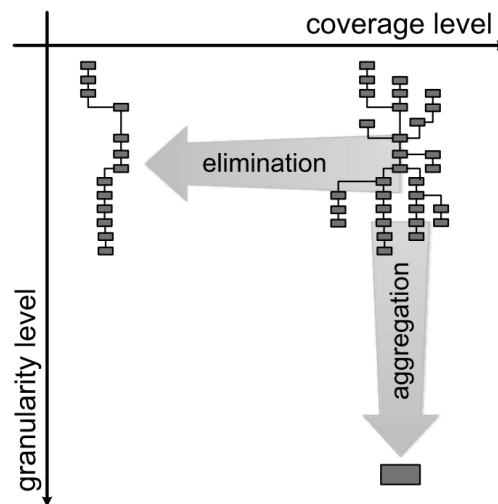


FIGURE 55: Distinction entre abstraction par élimination et abstraction par agrégation, repris de [81].

exécutées, et le deuxième vise à afficher les instances les plus fréquentes³. Le premier cas ne prend en compte que les activités les plus fréquentes indépendamment du chemin ; les autres activités sont alors éliminées du modèle. Pour éviter d'aboutir à un modèle inconsistant, les auteurs rajoutent une contrainte à ce cas d'utilisation : les activités peu fréquentes sont agrégées pour maintenir la consistance du modèle. Le troisième cas d'utilisation, qui se rapproche le plus de notre problématique, vise à proposer une vision simplifiée de tout le modèle en agrégeant les activités pour en simplifier la compréhension. Nous présentons dans le reste de cette section plusieurs travaux portant sur l'abstraction de processus métier permettant de répondre à ce cas d'utilisation.

Eshuis et al. [25] proposent une méthode pour construire des vues personnalisées de processus métier structurés [48]. L'objectif de leur proposition est de faciliter l'échange de processus métier entre clients et fournisseurs. Leur méthode comporte deux aspects : masquer certains éléments sensibles du savoir-faire du fournisseur et permettre au client d'obtenir une vue simplifiée du processus métier. Elle se déroule donc en deux étapes : dans un premier temps, le fournisseur choisit les activités qu'il souhaite agréger, puis le client choisit les activités qui l'intéresse ; les autres activités sont alors masquées. La Figure 56 illustre les deux étapes de leur proposition sur un modèle simple non structuré. Le modèle dans la partie basse de la figure est, dans un premier temps, simplifié par le fournisseur : les activités B et D ne doivent pas être présentées et sont donc agrégées avec l'activité C dans une activité abstraite BCD. Dans un deuxième temps, le client ne souhaite voir que les activités A et F. Les activités BCD et E sont conservées pour assurer la cohérence du modèle mais agrégées en une activité BCDE.

³. Une instance est un chemin d'activités, c'est-à-dire l'enchaînement des activités lors d'une exécution.

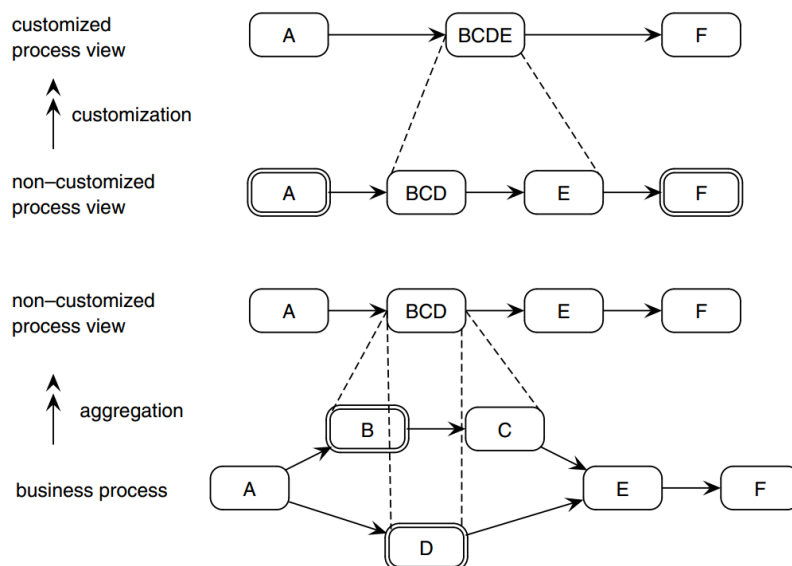


FIGURE 56: Illustration de la méthode d'abstraction de Eshuis et al [25].

Dans [8], Bobrik et al. proposent une méthode de simplification d'un processus métier en masquant les activités en fonction de l'acteur qui les réalise. Mais cette proposition ne s'arrête pas à la réduction du nombre d'activités du processus métier. Les auteurs proposent aussi de mettre en forme les différentes activités en fonction de l'acteur qui doit la réaliser, en fonction de leur état (en cours d'exécution, en attente, etc.) ou encore en fonction des données qui leur sont associées. Leur méthode d'abstraction d'un modèle de processus métier est présentée dans [9] et [10]. Elle consiste à supprimer ou agréger certaines activités et se base sur la notion de fragments à un seul point d'entrée et un seul point de sortie (reprenant ainsi la notion de Single Entry Single Exit - SESE - de la théorie des graphes [46]). La suppression d'activités (aussi appelé réduction) requiert une étape de simplification pour, par exemple, supprimer des branches ne comportant plus d'activité. L'agrégation d'activités est réalisée en fonction d'un ensemble de règles d'agrégation essentiellement utile pour l'agrégation d'activités parallèles (un fragment SESE est simplement agrégé en une nouvelle activité). La Figure 57 illustre les deux types d'abstraction de leur méthode pour un modèle de processus métier comportant deux fragments SESE tel que présenté dans la Figure 57a.

Smirnov et al. ont proposé une méthode d'abstraction basée sur l'agrégation des activités en fonction de leurs propriétés [80]. Les propriétés prises en compte sont, par exemple, les données manipulées par les activités ou les acteurs les réalisant. Pour agréger les activités similaires, une mesure de similarité pour chaque propriété entre chaque paire d'activités est calculée en utilisant la distance cosinus (empruntée à la recherche d'information). Ces mesures sont pondérées pour calculer une seule mesure entre chaque activité. Les pondérations sont précisées par les utilisateurs en fonction de leurs préférences.

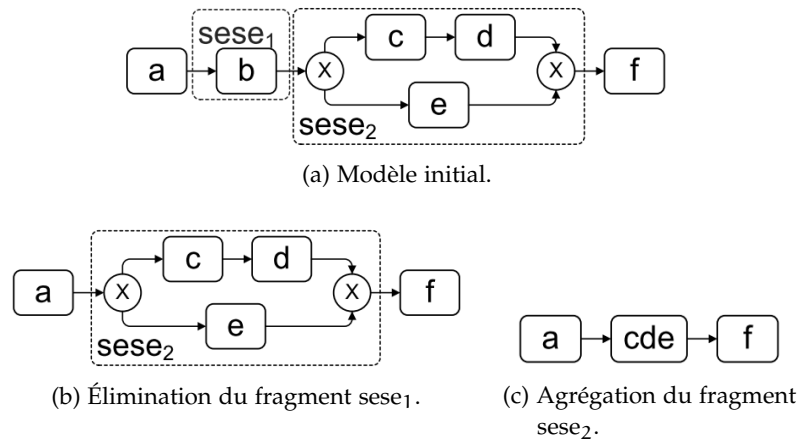


FIGURE 57: Illustration de la méthode d'abstraction de Bobrik et al. [8, 9, 10].

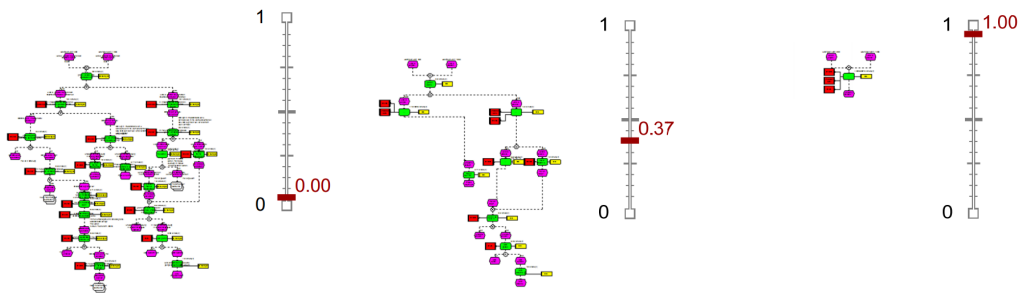


FIGURE 58: Illustration de la méthode d'abstraction de Polyvyanyy et al. [69].

Dans [69], Polyvyanyy et al. ont proposé une méthode d'abstraction interactive basée sur l'utilisation d'un *slider* qui permet à l'utilisateur de préciser le niveau d'abstraction qu'il souhaite obtenir. L'abstraction est réalisée en fonction d'un certain nombre de critères qui sont, par exemple, l'effort nécessaire à la réalisation d'une activité ou la probabilité de réalisation d'une activité dans une instance d'un modèle. La Figure 58 illustre l'utilisation d'un *slider* pour définir le niveau d'abstraction d'un modèle de processus métier.

Les règles de transformation qui sont appliquées ont été présentées dans [70] et, en fonction du critère d'abstraction, correspondent à une élimination ou une agrégation. Si le critère correspond à l'effort demandé par chaque activité, seules des opérations d'agrégation sont acceptables pour éviter de perdre de l'effort relatif aux activités éliminées. Les auteurs ont identifié quatre règles de transformation (voir Figure 59) : l'abstraction d'une suite séquentielle d'activités (voir Figure 59a), l'abstraction d'un block d'activités⁴ (voir Figure 59b), l'abstraction d'une boucle (voir Figure 59c), l'abstraction d'une impasse⁵ (voir Figure 59d).

Dans [37], Günther et al. ont étudié la simplification des modèles de processus métier générés à partir de l'étude des traces d'exécution. Ces traces sont

4. Un block d'activités est un ensemble d'activités contenues entre un connecteur AND et un connecteur JOIN.

5. Une impasse est un fragment du modèle qui n'est pas relié à son point de sortie.

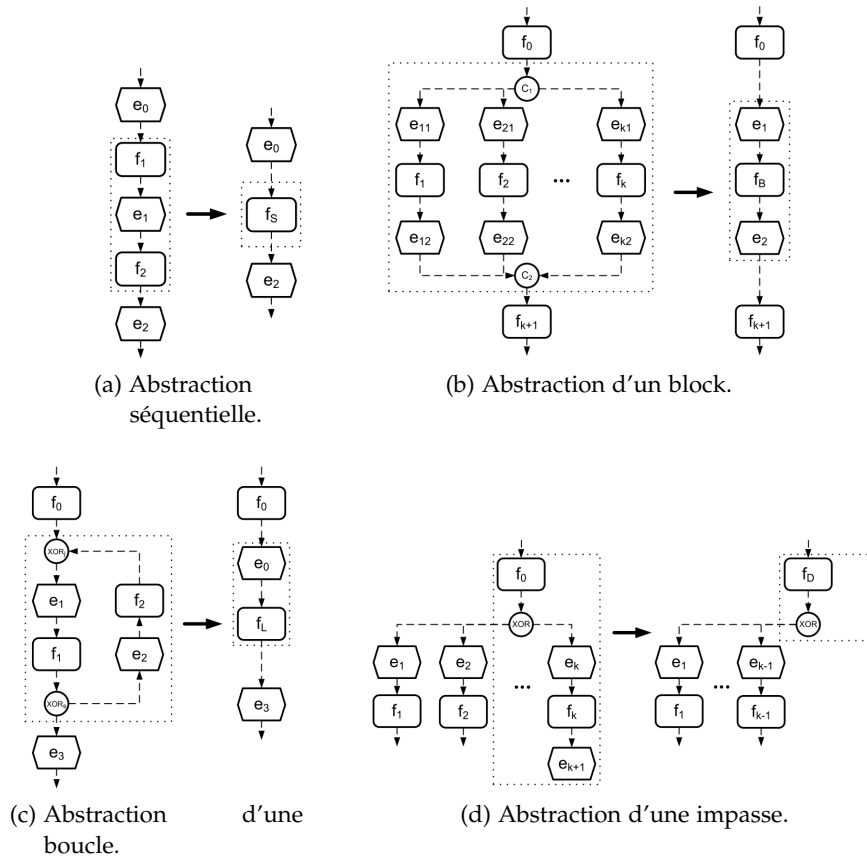


FIGURE 59: Règles d'abstraction de Polyvyanyy et al. [70].

issues d'applications de gestion de processus métier comme des ERP ou des CRM. Ces systèmes d'information enregistrent, entre autres, des informations sur les activités qui ont été réalisées, par qui et à quel moment. L'analyse de ces traces permet de réaliser plusieurs opérations automatiquement comme la découverte de modèles de processus métier [94]. Mais les processus obtenus comportent souvent trop de bruit pour être utilisables. La méthode de Günther et al. s'applique à un modèle de processus métier réduit à sa plus simple expression : un graphe orienté dont les nœuds sont des activités et les arcs des relations de dépendance. Leur méthode consiste d'abord à réduire le nombre d'arcs en se basant sur la fréquence de leurs occurrences dans les traces d'exécution, puis à agréger les activités. Leur méthode d'agrégation se base sur des métriques de corrélation entre deux activités et d'importance de la relation entre deux activités (basée sur la distance et la fréquence). La Figure 60 illustre cette méthode sur un processus métier découvert dans des traces d'exécution (60a). Les pondérations des arcs et des nœuds correspondent à leurs fréquences dans les traces d'exécution.

Il existe aussi un ensemble de travaux dont les objectifs ne sont pas l'abstraction de processus métier mais qui peuvent apporter des méthodes utiles pour celui-ci. Ces travaux ont pour point commun d'utiliser des méthodes de réduction ou de décomposition d'un modèle de processus métier. La réduction de

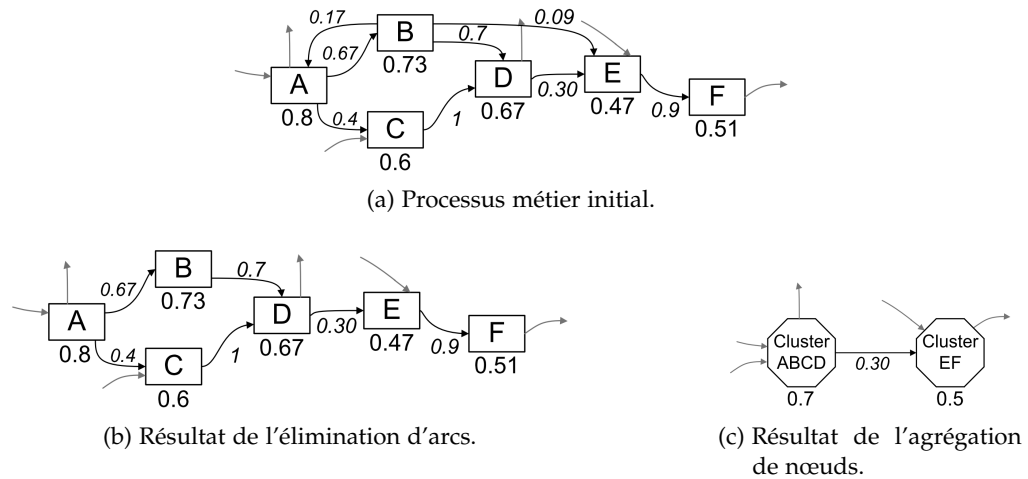


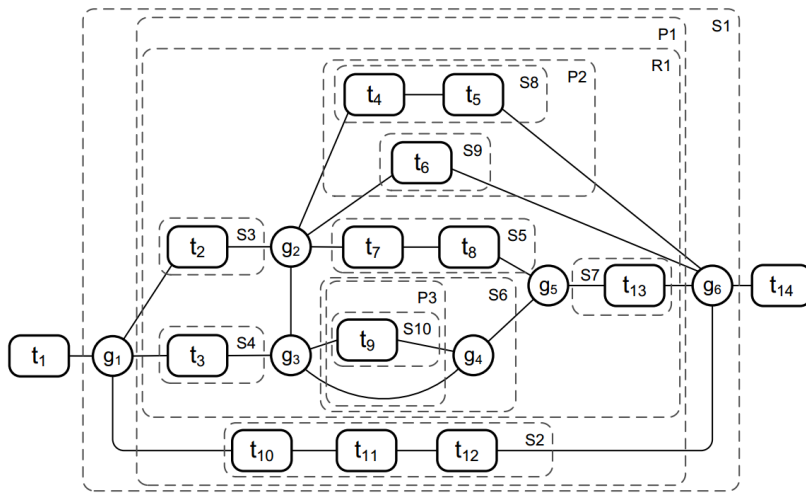
FIGURE 60: Illustration de la méthode d'abstraction de processus métier découverts dans les traces d'exécution proposée par Günther et al. dans [37].

processus métier s'inspire des travaux de Bertholet sur la réduction de réseau de Petri [6] dans lesquels un graphe est itérativement réduit à un seul nœud en appliquant des règles de réduction.

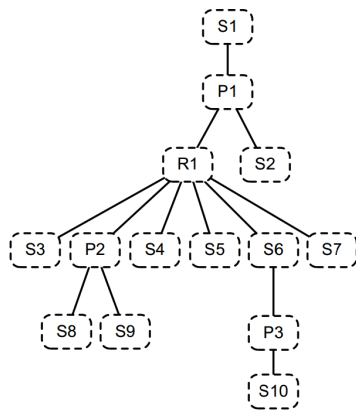
Dans [76] Sadiq et al. ont proposé une méthode de détection de conflits structurels⁶ dans un modèle de processus métier basé sur une méthode de réduction. Un modèle comportera des erreurs s'il comporte d'autres nœuds que le nœud d'entrée et le nœud de sortie. Cette méthode permet aussi de compter le nombre d'instances possibles à partir d'un modèle ce qui fournit une indication sur sa complexité. Mendeling et al. ont proposé dans [58] une méthode basée sur des règles de réduction pour analyser la justesse d'un modèle. Leur analyse a porté sur le référentiel de processus métier de SAP pour lequel ils ont mis en évidence un certain nombre d'erreurs parmi les 600 processus métier qu'il contient. Ils ont aussi émis des hypothèses sur la cause des erreurs de modélisation qui les ont ensuite amenés à effectuer les recherches sur la compréhensibilité des modèles de processus métier que nous avons présentées dans la deuxième section de ce chapitre.

Les méthodes de décomposition d'un modèle de processus permettent d'obtenir plusieurs fragments de processus. Vanhatalo et al. ont proposé dans [96, 97] une méthode de décomposition d'un modèle de processus métier reprenant le principe de la décomposition de graphes proposée par Tarjan et Valdes dans [86]. Cette méthode repose sur la détection des *tri-connected components* dans un graphe bi-connecté pour obtenir un arbre de la structure du processus. Bien que la motivation de ces travaux concerne la traduction d'un modèle exprimé avec BPMN vers un modèle exprimé en BPEL, cette méthode se prête aussi à l'abstraction de processus. C'est ce que Polyvyanyy et al. ont proposé dans [71] en précisant des règles d'abstraction en fonction de l'arbre de la structure du

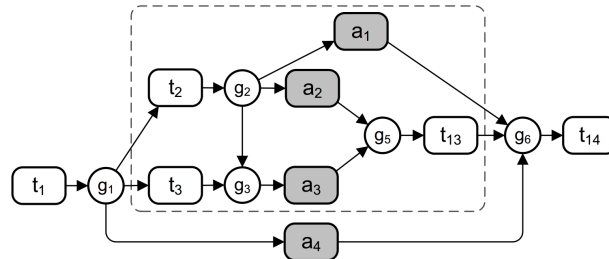
6. Les conflits structurels considérés dans ces travaux sont des cas d'inter-blocage ou d'absence de synchronisation entre des activités.



(a) Processus métier d'origine avec ses composantes *tri-connected*.



(b) Arbre de ses composantes *tri-connected*.



(c) Abstraction du processus d'origine, les activités en gris correspondent à des activités agrégées.

FIGURE 61: Illustration de la méthode d'abstraction basée sur la décomposition de processus métier proposée par Polyvyany et al. [71].

processus. La Figure 61 illustre cette méthode. Notons que dans cette méthode les processus qui sont agrégés sont préalablement choisis.

8.4 SYNTHÈSE

Bien que la modélisation de processus métier soit plus complexe que la modélisation des flux économiques de l'ACV, on retrouve les problématiques de compréhension et de maintenabilité que nous avons choisies de traiter dans cette thèse. La modularité présente dans les langages de modélisation apporte une solution à cette problématique mais elle ne permet pas d'obtenir une traduction entre un modèle abstrait et un modèle détaillé. Les méthodes d'abstraction que nous avons présentées permettent de réaliser cette traduction automatique, mais elles reposent sur une approche du bas vers le haut, là où notre proposition propose une approche du haut vers le bas. De plus, notre méthode de modélisation permet d'agréger les processus de l'ACV sémantiquement proches plutôt que les séquences de processus et d'exprimer des relations de dépendance entre des processus agrégés. Mais les méthodes d'abstraction de processus métier pourront se révéler utiles pour générer un macro-graphe à partir d'une base d'inventaires existante.

CONCLUSION

Depuis plus de trente ans, les spécialistes de l'environnement nous alertent sur les conséquences des pollutions générées par l'Homme. L'une des méthodes les plus précises et complètes pouvant contribuer à l'évaluation, au contrôle, et à la réduction des rejets émis par les activités humaines est l'analyse en cycles de vie. Dans cette thèse nous avons abordé deux problématiques liées à l'ACV. La première concerne la compréhensibilité et la maintenabilité des bases d'inventaires en cycle de vie. La seconde concerne l'optimisation du calcul des flux cumulés et la détermination des raisons de non calculabilité d'un inventaire des flux économiques.

Pour aborder la première problématique nous avons proposé une méthode sémantique de gestion d'une base de données d'inventaires en cycle de vie. Pour la seconde nous avons proposé d'utiliser une méthode de décomposition des flux économiques d'un inventaire.

9.1 CONTRIBUTIONS

Notre principale contribution est une **méthode de gestion d'une base de données d'inventaires basée sur la sémantique des processus** qu'elle contient. Cette méthode repose sur l'indexation sémantique des processus et des coefficients et sur le regroupement de ces éléments dans des groupes multi dimensionnels. Nous avons proposé d'utiliser deux niveaux de détails : un pour les relations entre des ensembles de processus et un pour les relations entre les processus individuels. Notre proposition permet d'augmenter la compréhensibilité d'une base d'inventaires en proposant une vue simplifiée des relations de dépendance à l'utilisateur. Elle permet aussi d'augmenter sa maintenabilité en diminuant le nombre de relations de dépendance à maintenir. La Figure 62 illustre les deux niveaux de notre approche.

Nous avons proposé une **formalisation** de cette méthode ainsi que les **algorithmes de transformation** des relations du macro-graphe en relations du graphe détaillé. Pour assurer l'efficacité de ces algorithmes, nous avons étudié leurs **passages à l'échelle**. Pour faciliter l'utilisation de cette méthode, nous avons proposé une **approche déclarative utilisant l'algèbre relationnelle** et SQL. Et, pour mettre à profit l'indexation sémantique des éléments de la base d'inventaires, nous avons proposé une **approche déclarative basée sur les outils du web sémantique** permettant d'analyser la chaîne de composition des processus.

La méthode a été testée dans le cadre d'une **étude de cas** sur une base d'inventaires existante et a été implémentée dans un **prototype : CarbonDB**. Dans ce prototype, nous avons aussi proposé une interface utilisateur construite au-

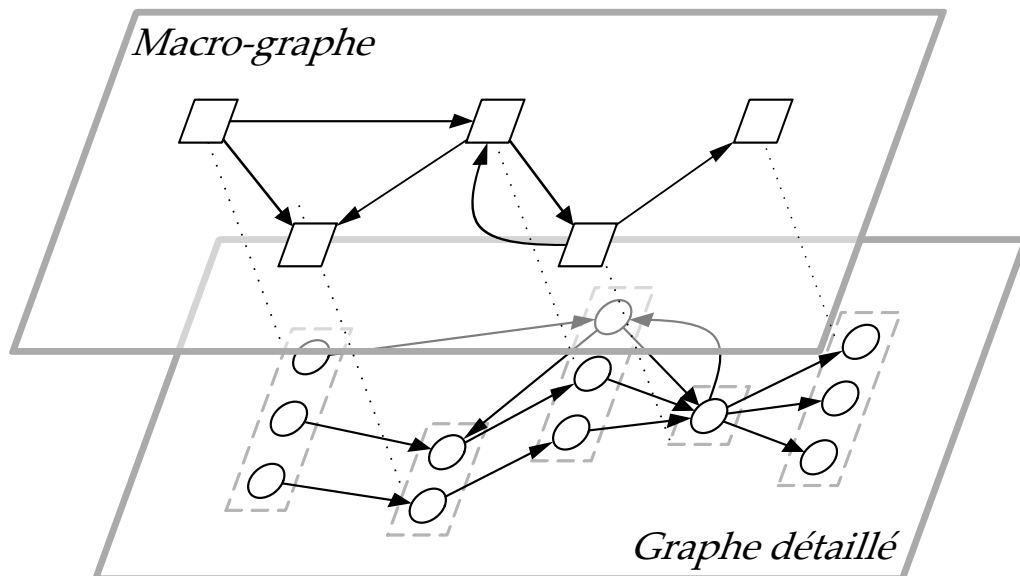


FIGURE 62: Illustration de la modélisation à deux niveaux d'une base d'inventaires en cycle de vie.

tour de la navigation au sein du macro-graphe et du graphe détaillé. My C-Sense a repris l'indexation sémantique et le regroupement sémantique dans une **application opérationnelle**. La modélisation à deux niveaux des relations de dépendance sera reprise quand les données fournies par l'ADEME incluront ce type d'informations ou quand My C-Sense sera amenée à fournir des données issues d'autres sources à ses clients.

L'inconvénient de cette méthode tient à la maintenance de l'ontologie des mots-clés servant à décrire les processus et coefficients. Cette opération peut se révéler complexe pour un utilisateur non-initié à la modélisation d'ontologie. C'est l'un des points que nous souhaitons aborder dans de futurs travaux.

Nous avons réalisé un **état des lieux des méthodes de calcul des flux cumulés** de l'ACV pour déterminer la méthode la plus efficace pour les calculer. Puis, nous avons proposé de réaliser une **triangularisation par bloc de la matrice des flux** pour :

- déterminer quelle sous-partie du modèle n'est pas calculable ;
- optimiser le calcul des flux dans un environnement dynamique.

Mais ce découpage par bloc est limité par la topologie du graphe détaillé, qui n'est pas décomposable s'il ne comporte qu'une seule composante fortement connexe.

9.2 PERSPECTIVES

9.2.1 *Génération automatique du macro-graphe*

Pour simplifier le travail de l'éditeur d'une base d'inventaires intégrant notre proposition, nous souhaitons expérimenter l'utilisation de ressources sémantiques existantes. L'étude sera axée sur l'utilisation d'une ontologie généraliste ou d'un ensemble d'ontologies de domaines pour indexer les éléments de la base. Ceci pourrait servir de point de départ à la construction automatique d'un macro-graphe à partir d'une base d'inventaires déjà existante. Pour atteindre cet objectif, nous serons amenés à utiliser des techniques d'extraction d'entités nommées dans les métadonnées des processus, et des méthodes de clustering de graphes [78] pour regrouper les processus sémantiquement proches. Cette évolution se rapproche des méthodes d'abstraction de processus métier que nous avons présentées dans le chapitre **BUSINESS PROCESS MODELING**.

Nous souhaitons aussi étudier l'intérêt d'étendre notre modélisation à n niveaux en créant des ensembles de groupes de processus. Nous envisageons d'étudier cette évolution après avoir réalisé la génération automatique du macro-graphe. Le prototype pourrait alors inclure une interface de navigation entre les différents niveaux d'abstraction.

9.2.2 *Analyse en cycle de vie hybride*

Notre approche pourrait être utile pour l'ACV hybride mêlant l'ACV traditionnel et l'analyse E/S environnementale. Les groupes de processus pourraient être associés à des flux vers l'environnement correspondant à un processus générique pour l'ensemble des processus du groupe. Considérons un groupe des processus de production d'électricité aux États-Unis avec une dimension contenant tous les états américains. Supposons que nous ne possédions pas le détail pour un des états et qu'un processus aval consomme de l'électricité provenant de plusieurs états y compris de celui-ci. Dans ce cas, nous pourrions créer une relation de dépendance entre le groupe de production d'électricité et ce processus. La relation détaillée créée pour l'état sans détail utiliserait le processus générique de production d'électricité sur tout le territoire américain. Nous souhaitons donc conduire une étude de cas en utilisant les données d'une base d'inventaires d'ACV et les données d'une base d'analyse E/S environnementale.

9.2.3 *Calcul des flux cumulés*

La décomposition du graphe détaillé est utile pour orienter l'utilisateur vers la partie du modèle qui rend l'ensemble non calculable. Mais cette méthode n'est utile que si nous avons plusieurs composantes fortement connexes dans le graphe détaillé. Aussi, nous allons essayer de trouver des méthodes fournissant ce type d'informations dans un graphe détaillé ne comportant qu'une seule composante fortement connexe.

BIBLIOGRAPHIE

- [1] ADEME. Guide des facteurs d'émissions. Technical report, Agence de l'Environnement et de la Maîtrise de l'Energie, 2007. (Cité page 18.)
- [2] Chris Aitken, Christine Stephenson, and Ryan Brinkworth. Process Classification Frameworks. In *Handbook on Business Process Management 2*, pages 73–92. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-01981-4. (Cité page 99.)
- [3] Edward Anderson. *LAPACK Users' Guide*. Siam, 1999. ISBN 0898714478. (Cité page 87.)
- [4] Charlton Barreto, Vaughn Bullard, Thomas Erl, John Evdemon, Diane Jordan, Khanderao Kand, Dieter König, Simon Moser, Ralph Stout, Ron Ten-Hove, Ivana Trickovic, Danny van der Rijn, and Alex Yiu. Web services business process execution language version 2.0 (OASIS standard). WS-BPEL. Technical report, OASIS Open, 2007. (Cité page 98.)
- [5] David Beckett and Tim Berners-Lee. Turtle Terse RDF Triple Language. Technical report, W3C, 2011. URL www.w3.org/TR/turtle/. (Cité page 68.)
- [6] Gérard Berthelot. Checking properties of nets using transformations. In *Advances in Petri Nets 1985*, pages 19–40. Springer Berlin Heidelberg, 1986. ISBN 978-3-540-16480-7. (Cité page 106.)
- [7] Benjamin Bertin, Vasile-Marian Scuturici, Jean-Marie Pinon, and Emmanuel Risler. CarbonDB : a semantic life cycle inventory database. In *CIKM'12 Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2683–2685. ACM, 2012. ISBN 9781450311564. (Cité page 89.)
- [8] Ralph Bobrik, Thomas Bauer, and Manfred Reichert. Proviado—personalized and configurable visualizations of business processes. *E-Commerce and Web Technologies*, pages 61–71, 2006. (Cité pages xviii, 103 et 104.)
- [9] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. View-based process visualization. *Business Process Management*, pages 88–95, 2007. (Cité pages xviii, 103 et 104.)
- [10] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. Parameterizable views for process visualization. Technical report, Centre for Telematics and Information Technology, University of Twente, 2007. (Cité pages xviii, 103 et 104.)
- [11] Michael Bostock and Jeffrey Heer. Protovis : a graphical toolkit for visualization. *IEEE transactions on visualization and computer graphics*, 15(6) : 1121–1128, 2009. ISSN 1077-2626. (Cité page 89.)

- [12] Richard A. Brualdi and Herbert J. Ryser. *Combinatorial Matrix Theory*. Cambridge University Press, 1991. ISBN 0521322650. (Cité page 82.)
- [13] Carnegie Mellon University Green Design Institute. Economic Input-Output Life Cycle Assessment (EIO-LCA), 2008. URL <http://www.eiolca.net>. (Cité page 17.)
- [14] Tzu-Yi Chen. *Preconditioning sparse matrices for computing eigenvalues and solving linear systems of equations*. PhD thesis, University of California at Berkeley, 2001. (Cité page 83.)
- [15] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3) :251–280, 1990. ISSN 07477171. (Cité page 80.)
- [16] Thomas H. Cormen. *Introduction to Algorithms*. Mit Press, 2009. ISBN 0262533057. (Cité page 80.)
- [17] Biswa Nath Datta. *Numerical Linear Algebra and Applications*. SIAM, 2010. ISBN 0898716853. (Cité page 79.)
- [18] Thomas H. Davenport. *Process Innovation : Reengineering Work Through Information Technology*. Harvard Business Press, 1993. ISBN 1422160661. (Cité page 97.)
- [19] Chris Davis. *Integration of life cycle analysis within agent based modeling using a case study on bio-electricity*. Master's thesis, Delft University of Technology, 2007. (Cité page 13.)
- [20] Chris Davis and Igor Nikoli. Integration of Life Cycle Assessment Into Agent-Based Modeling Infrastructure Systems. *Journal of Industrial Ecology*, 13(2), 2009. doi : 10.1111/j.1530-9290.2009.00122.x. (Cité page 13.)
- [21] Timothy A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006. ISBN 0898716136. (Cité page 80.)
- [22] Timothy A Davis and Hu Yifan. The University of Florida sparse matrix collection. *ACM transactions on mathematical software*, 38 :28, 2012. (Cité page 87.)
- [23] Iain S. Duff and John K. Reid. An Implementation of Tarjan's Algorithm for the Block Triangularization of a Matrix. *ACM Transactions on Mathematical Software*, 4(2) :137–147, 1978. ISSN 00983500. (Cité page 83.)
- [24] Andrew L. Dulmage and Nathan S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10(4) :516–534, 1958. (Cité page 83.)
- [25] Rik Eshuis and Paul Grefen. Constructing customized process views. *Data & Knowledge Engineering*, 64(2) :419–438, 2008. ISSN 0169023X. (Cité pages xviii, 102 et 103.)

- [26] European Commission Joint Research Centre. European reference Life-Cycle Database, 2012. URL <http://elcd.jrc.ec.europa.eu>. (Cité page 13.)
- [27] Rolf Frischknecht. Life cycle inventory modelling in the Swiss national database ECOINVENT 2000. *Sustainability in the Information Society*, 2 :24, 2001. (Cité page 13.)
- [28] Rolf Frischknecht and Niels Jungbluth. The ecoinvent database : Overview and methodological framework. *The International Journal of Life Cycle Assessment*, 10(1) :3–9, 2005. (Cité pages 3, 13 et 43.)
- [29] Rolf Frischknecht and Gerald Rebitzer. The ecoinvent database system : a comprehensive web-based LCA database. *Journal of Cleaner Production*, 13(13-14) :1337–1343, 2005. ISSN 09596526. (Cité pages 13 et 81.)
- [30] GaBi. Gabi Life Cycle Inventory Databases, 2011. URL <http://www.gabi-software.com>. (Cité page 3.)
- [31] Abdul Azim Abdul Ghani, Koh tieng Wei, Geoffrey Muchiri Muketha, and Wong Pei Wen. Complexity metrics for measuring the understandability and maintainability of business process models using goal-question-metric (GQM). *International Journal of Computer Science and Network Security*, 8(5) :219–225, 2008. (Cité pages 100 et 101.)
- [32] Aditya Ghose, Konstantin Hoesch-Klohe, Lothar Hinsche, and Lam-Son Le. Green business process management : A research agenda. *Australasian Journal of Informations Systems*, 16(2) :103–117, 2010. (Cité page 99.)
- [33] Anne Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, 1987. ISBN 089871396X. (Cité page 80.)
- [34] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220, 1993. (Cité page 24.)
- [35] Volker Gruhn and Ralf Laue. Complexity metrics for business process models. In *9th international conference on business information systems (BIS 2006)*, pages 1–12, 2006. (Cité page 101.)
- [36] Jereon Guinée. *Handbook on Life Cycle Assessment : Operational Guide to the ISO Standards*. Springer, 2002. ISBN 1402002289. (Cité pages 9 et 22.)
- [37] Christian W. Günther and Wil M. P. van der Aalst. Fuzzy mining–adaptive process simplification based on multi-perspective metrics. *Business Process Management*, pages 328–343, 2007. (Cité pages xviii, 104 et 106.)
- [38] Wolfgang Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer, 2012. ISBN 1461287243. (Cité page 80.)
- [39] Michael Hammer and James Champy. *Reengineering the Corporation : A Manifesto for Business Revolution*. HarperCollins, 2003. ISBN 0060559535. (Cité page 97.)

- [40] Pat Hayes and Chris Welty. Defining N-ary Relations on the Semantic Web. Technical report, W3C, 2006. URL www.w3.org/TR/swbp-n-aryRelations/. (Cité page 72.)
- [41] Jan Hedemann, Ingo Meinshausen, and Rolf Frischknecht. Documentation EcoSpold. Technical report, Swiss Centre for Life Cycle Inventories, 2008. (Cité page 51.)
- [42] Chris Hendrickson, Arpad Horvath, Satish Joshi, and Lester Lave. Economic Input-Output Models for Environmental Life-Cycle Assessment. *Environmental science & technology*, 32(7) :184–191, 1998. ISSN 1520-5851. (Cité page 16.)
- [43] Ivan Herman, Guy Melançon, and M. Scott Marshall. Graph visualization and navigation in information visualization : A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1) :24–43, 2000. ISSN 10772626. doi : 10.1109/2945.841119. (Cité page 90.)
- [44] Konstantin Hoesch-Klohe, Aditya Ghose, and Lam-Son Lê. Towards green business process management. In *Services Computing (SCC), 2010 IEEE International Conference on*, pages 386–393. IEEE, 2010. (Cité page 99.)
- [45] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosf, and Mike Dean. SWRL : A Semantic Web Rule Language Combining OWL and RuleML. Technical report, W3C, 2004. (Cité page 73.)
- [46] Richard Johnson, David Pearson, and K Pingali. The program structure tree : Computing control regions in linear time. *ACM SigPlan Notices*, (June), 1994. (Cité page 103.)
- [47] Gerhard Keller and Thomas Teufel. *Sap R/3 Process Oriented Implementation*. August 1998. ISBN 0201924706. (Cité page 98.)
- [48] Bartek Kiepuszewski, A ter Hofstede, and C Bussler. On structured workflow modelling. *Advanced Information Systems Engineering*, pages 431–445, 2000. (Cité page 102.)
- [49] Graham Klyne and Jeremy Carroll. Resource Description Framework (RDF) : Concepts and Abstract Syntax. Technical Report W3C Recommendation 10 February 2004, W3C, 2004. URL www.w3.org/RDF/. (Cité page 71.)
- [50] Ryan K.L. Ko, Stephen S.G. Lee, and Eng Wah Lee. Business process management (BPM) standards : a survey. *Business Process Management Journal*, 15(5) :744–791, 2009. ISSN 1463-7154. (Cité pages xviii, 97 et 98.)
- [51] M. Laydi. Calcul matriciel. (Cité page 79.)
- [52] Wassily Leontief. Environmental Repercussions and the Economic Structure : An Input-Output Approach. *The Review of Economics and Statistics*, 52(3) :262–271, 1970. (Cité page 16.)

- [53] Deborah L. McGuinness. Ontologies come of age. *Spinning the semantic web : bringing the World Wide Web to its full potential*, pages 171–192, 2002. (Cité pages 24 et 89.)
- [54] Deborah L. McGuinness and Frank Van Harmelen. OWL web ontology language overview. Technical report, W3C recommendation, 2004. URL <http://www.w3.org/TR/owl-ref/>. (Cité page 57.)
- [55] Jim Melton and Alan R. Simon. *SQL : 1999 - Understanding Relational Language Components*. Morgan Kaufmann, 2001. ISBN 0080517609. (Cité pages 57 et 67.)
- [56] Jan Mendling, Gustaf Neumann, and Wil M. P. van der Aalst. Understanding the occurrence of errors in process models based on metrics. *On the Move to Meaningful Internet Systems 2007 : CoopIS, DOA, ODBASE, GADA, and IS*, pages 113–130, 2007. (Cité page 100.)
- [57] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What makes process models understandable? *Business Process Management*, pages 48–63, 2007. (Cité page 100.)
- [58] Jan Mendling, H Verbeek, B.F. van Dongen, Wil M. P. van der Aalst, and Gustaf Neumann. Detection and prediction of errors in EPCs of the SAP reference model. *Data & Knowledge Engineering*, 64(1) :312–329, January 2008. ISSN 0169023X. (Cité page 106.)
- [59] Jan Mendling, Hajo A. Reijers, and Wil M. P. van der Aalst. Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52(2) : 127–136, February 2010. ISSN 09505849. (Cité page 100.)
- [60] Hafedh Mili, Guy Tremblay, Guitta Bou Jaoude, Éric Lefebvre, Lamia Elabed, and Ghizlane El Boussaidi. Business process modeling languages. *ACM Computing Surveys*, 43(1) :1–56, November 2010. ISSN 03600300. (Cité page 98.)
- [61] PH Nielsen, AM Nielsen, BP Weidema, R Dalgaard, and N Halberg. LCA food data base, 2003. URL <http://www.lcafood.dk>. (Cité page 13.)
- [62] Alexander Nowak, Frank Leymann, and Ralph Mietzner. Towards green business process reengineering. *Service-Oriented Computing*, pages 187–192, 2011. (Cité page 99.)
- [63] Alexander Nowak, Frank Leymann, David Schumm, and Branimir Wetzstein. An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering. *Information and Communication on Technology for the Fight against Global Warming*, pages 150–164, 2011. (Cité page 99.)
- [64] NREL. U.S. Life Cycle Inventory Database, 2011. URL <http://www.nrel.gov/lci/>. (Cité pages 3, 13, 49 et 51.)

- [65] Martin O'Connor and Amar Das. SQWRL : a Query Language for OWL. In *Proc. of 6th OWL : Experiences and Directions Workshop (OWLED2009)*, page 8, 2009. (Cité page 73.)
- [66] OMG. Business Process Model And Notation (BPMN). Technical Report January, Object Management Group, 2011. (Cité page 98.)
- [67] Glen P. Peters. Efficient Algorithms for Life Cycle Assessment, Input-Output Analysis, and Monte-Carlo Analysis. *Science And Technology*, 12 (6) :373 – 380, 2007. (Cité page 81.)
- [68] Carl A. Petri. *Communication with Automata*. PhD thesis, Universität Bonn, 1962. (Cité page 98.)
- [69] Artem Polyvyanyy, Sergey Smirnov, and Mathias Weske. Process model abstraction : A slider approach. *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE*, pages 325–331, 2008. (Cité pages xviii et 104.)
- [70] Artem Polyvyanyy, Sergey Smirnov, and Mathias Weske. Reducing complexity of large EPCs. *MobIS 2008*, pages 195–207, 2008. (Cité pages xviii, 104 et 105.)
- [71] Artem Polyvyanyy, Sergey Smirnov, and Mathias Weske. The triconnected abstraction of process models. *Business Process Management*, pages 229–244, 2009. (Cité pages xviii, 106 et 107.)
- [72] Alex Pothén and Chin-Ju Fan. Computing the block triangular form of a sparse matrix. *ACM Transactions on Mathematical Software*, 16(4) :303–324, 1990. ISSN 00983500. (Cité page 83.)
- [73] Eric Prud'Hommeaux and Andy Seaborne. SPARQL query language for RDF. Technical report, W3C, 2008. URL www.w3.org/TR/rdf-sparql-query/. (Cité pages 57 et 69.)
- [74] Hajo A. Reijers and Jan Mendling. Modularity in process models : Review and effects. *Business Process Management*, pages 20–35, 2008. (Cité page 100.)
- [75] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003. ISBN 0898715342. (Cité page 80.)
- [76] Wasim Sadiq and Maria E. Orlowska. On business process model transformations. *Conceptual Modeling - ER 2000*, pages 267–280, 2000. (Cité page 106.)
- [77] Kamyar Sarshar and Peter Loos. Comparing the Control-Flow of EPC and Petri Net from the End-User Perspective. *Business Process Management*, pages 434–439, 2005. (Cité page 100.)
- [78] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1) :27–64, August 2007. ISSN 15740137. doi : 10.1016/j.cosrev.2007.05.001. (Cité page 111.)

- [79] Sergey Smirnov, Hajo A. Reijers, Thijs Nugteren, and Mathias Weske. *Business process model abstraction : theory and practice*. 2010. ISBN 3869560541. (Cité page 101.)
- [80] Sergey Smirnov, Hajo A. Reijers, and Mathias Weske. A semantic approach for business process model abstraction. *Advanced Information Systems Engineering*, pages 497–511, 2011. (Cité page 103.)
- [81] Sergey Smirnov, Hajo A. Reijers, Mathias Weske, and Thijs Nugteren. Business process model abstraction : a definition, catalog, and survey. *Distributed and Parallel Databases*, 30(1) :63–99, January 2012. ISSN 0926-8782. (Cité pages xviii, 101 et 102.)
- [82] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4) :354–356, August 1969. ISSN 0029-599X. (Cité page 80.)
- [83] Sangwon Suh and Gjalte Huppel. Missing inventory estimation tool using extended input-output analysis. *The International Journal of Life Cycle Assessment*, 7(3) :134–140, May 2002. ISSN 0948-3349. (Cité page 19.)
- [84] Sangwon Suh and Shinichiro Nakamura. Five years in the area of input-output and hybrid LCA. *The International Journal of Life Cycle Assessment*, 12(6) :351–352, 2007. (Cité page 19.)
- [85] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, pages 114–121, 1972. (Cité page 83.)
- [86] Robert Tarjan and Jacobo Valdes. Prime subprogram parsing of a program. In *POPL '80 Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 95–105, 1980. ISBN 0897910117. (Cité page 106.)
- [87] The International Organization for Standardization. ISO 14048 : Environmental management – Life cycle assessment – Data documentation format, 2002. (Cité page 16.)
- [88] The International Organization for Standardization. ISO 14040 : Environmental management - Life cycle assessment - Principles and framework, 2006. (Cité page 10.)
- [89] The International Organization for Standardization. ISO 14044 : Environmental management - Life cycle assessment - Requirements and guidelines, 2006. (Cité pages 1, 2 et 10.)
- [90] Lloyd N. Trefethen and David III Bau. *Numerical Linear Algebra*. SIAM, 1997. ISBN 0898713617. (Cité pages 80 et 81.)
- [91] Wil M. P. van der Aalst. *The Application of Petri Nets To Workflow Management*, 1998. ISSN 0218-1266. (Cité page 98.)
- [92] Wil M. P. van der Aalst. *Business Process Management : A Comprehensive Survey*. *ISRN Software Engineering*, 2013 :1–37, 2013. ISSN 2090-7680. (Cité pages xviii, 98 et 99.)

- [93] Wil M. P. van der Aalst, Arthur H. M. Hofstede, and Mathias Weske. Business Process Management : A Survey. pages 1–12, 2003. (Cité page 97.)
- [94] Wil M. P. van der Aalst, Tom Weijters, and Laura Maruster. Workflow mining : discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9) :1128–1142, September 2004. ISSN 1041-4347. (Cité page 105.)
- [95] Dieter Van Nuffel and Manu De Backer. Multi-abstraction layered business process modeling. *Computers in Industry*, 63(2) :131–147, February 2012. ISSN 01663615. (Cité page 99.)
- [96] Jussi Vanhatalo, H Völzer, and Frank Leymann. Faster and more focused control-flow analysis for business process models through SESE decomposition. *Service-Oriented Computing – ICSOC 2007*, pages 43–55, 2007. (Cité page 106.)
- [97] Jussi Vanhatalo, Hagen Völzer, and Jana Koehler. The refined process structure tree. *Business Process Management*, pages 100–115, 2008. (Cité page 106.)
- [98] Richard S. Varga. *Matrix Iterative Analysis*. Springer, 2009. ISBN 3642051545. (Cité page 79.)
- [99] David S. Watkins. *Fundamentals of Matrix Computations*. John Wiley & Sons, 2004. ISBN 0471461679. (Cité page 80.)
- [100] Raphael Yuster and Uri Zwick. Fast sparse matrix multiplication. *ACM Transactions on Algorithms*, 1(1) :2–13, July 2005. ISSN 15496325. (Cité page 82.)