Base de données sémantique d'inventaires en cycle de vie

Benjamin Bertin [1], **Vasile-Marian Scuturici** [1], **Jean-Marie Pinon** [1], **Emmanuel Risler** [2]

1. *Université de Lyon, CNRS*
   INSA-Lyon, LIRIS, UMR5205, F-69621, Villeurbanne cedex
   {benjamin.bertin,marian.scuturici,jean-marie.pinon}@liris.cnrs.fr
2. *Université de Lyon, CNRS*
   INSA-Lyon, ICJ, UMR5208, My C-Sense, F-69621, Villeurbanne cedex
   emmanuel.risler@insa-lyon.fr

RÉSUMÉ. *L'analyse en cycle de vie fournit une méthode reconnue pour la modélisation des impacts environnementaux d'un produit ou d'un service. Cette méthode se base sur la décomposition du système étudié en activités interdépendantes pour obtenir un inventaire en cycle de vie. La méthode usuelle pour gérer des bases de données d'inventaires, pouvant contenir plusieurs milliers d'activités, repose sur la manipulation d'éléments individuels. L'édition d'une telle base est souvent fastidieuse et peut être source d'erreurs. Nous proposons une nouvelle approche pour la gestion de ces bases qui repose sur la sémantique des activités. Cette approche facilite la modélisation des relations de dépendance entre les activités et offre une vue d'ensemble de celles-ci. Dans cet article, nous expliquons notre approche et des éléments-clefs de son implémentation. Nous présentons aussi une étude de cas basée sur la production d'électricité aux États-Unis et une expérimentation sur le passage à l'échelle de notre implémentation.*

ABSTRACT. *Life Cycle Assessment provides a well-accepted methodology for modelling environmental impacts of products and services. This methodology relies on the decomposition of a studied system into interdependent activities. The usual work-flow to manage activities databases, containing several thousands of elements, is based on the manipulation of individual items which turns out to be a very harnessing work and error prone. We propose a new work-flow for activities databases maintenance based on the addition of semantic information to the elements they contained. This method eases considerably the modelling process and also offers a more understandable model of the dependencies links. In this paper, we explain our approach and some key parts of the implementation. We also present a case study based on the U.S. electricity production and an experiment on the scalability of our implementation.*

MOTS-CLÉS : *analyse en cycle de vie, ontologie, environnement.*

KEYWORDS: *life cycle assessment, ontology, environment.*

DOI:10.3166/ISI.18.2.103-128 © 2013 Lavoisier

1. Introduction

La réduction des impacts environnementaux des activités humaines, tels que les émissions de gaz à effet de serre, nécessite de modéliser et évaluer les effets sur l'environnement de ces activités. C'est l'objectif de la méthode dite de l'analyse en cycle de vie (ACV) (ISO, 2006). Elle permet de déterminer les impacts liés à la production d'un produit, à un service ou, plus généralement, à toute activité humaine. Cette méthode peut prendre en compte toutes les étapes de la vie d'un produit comme la production ou le recyclage. L'ACV permet d'évaluer différents impacts tels que les émissions de gaz à effet de serre ou les rejets de produits chimiques.

Une étude basée sur la méthode ACV se décompose en quatre étapes (ISO, 2006). La première consiste à définir les objectifs et le champ d'étude. Pour 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 la production d'énergie, l'épandage d'engrais, un déplacement en avion, 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 *processus* ou *processus unitaires* [1] (ISO, 2006). Dans la suite de cet article, nous utiliserons le terme de processus pour référer à la notion de processus unitaire de l'ACV. La méthode consacrée pour modéliser les interactions entre les processus d'un inventaire et entre les processus et l'environnement, est basée sur l'utilisation d'une matrice d'Entrée/Sortie (Leontief, 1986).

Plusieurs organisations proposent des bases de données d'inventaires en cycle de vie (Frischknecht, 2005b ; GaBi, 2011 ; NREL, 2011) qui servent de base à l'analyse en cycle de vie d'un système. Mais ces bases de données peuvent contenir plusieurs milliers de processus, ce qui les rend difficiles à comprendre à moins de réaliser une étude détaillée de toute la base. Il existe pourtant des similarités sémantiques entre les processus et leurs relations, telles 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. Ces proximités sémantiques sont évidemment difficiles à appréhender quand elles sont dispersées dans une matrice d'Entrée/Sortie. De plus, la maintenance d'une base de données d'inventaires est une activité fastidieuse si on doit maintenir des relations de dépendance sémantiquement proches.

Nous proposons une nouvelle méthode pour modéliser une base de données d'inventaires reposant sur la sémantique des processus. Dans notre approche les processus sont sémantiquement indexés et nous utilisons cette indexation pour regrouper sémantiquement les processus. Ces regroupements sont ensuite utilisés pour créer des

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.

relations de dépendance entre plusieurs processus au lieu de créer des relations de dépendance entre des processus individuels. Avec cette modélisation nous répondons aux deux problématiques que nous avons identifiées : nous offrons une vue d'ensemble des données et nous facilitons la gestion des relations de dépendance. 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. 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 dans le graphe détaillé afin de calculer les impacts des processus.

La deuxième section de cet article présente succinctement la réalisation d'un inventaire en cycle de vie. Les troisième et quatrième sections présentent notre approche et sa formalisation en détail. Puis, nous expliquons dans les cinquième et sixième sections la méthode permettant de convertir des relations de dépendance entre des groupes de processus en relations de dépendance entre des processus individuels et son implémentation. Dans la septième section nous présentons les résultats d'une expérimentation sur le passage à l'échelle de l'algorithme de conversion. La dernière section est consacrée à une application de notre méthode sur des données extraites à partir de la base de données d'inventaires en cycle de vie du National Renewable Energy Laboratory (NREL, 2011). L'extrait que nous avons choisi concerne la production d'électricité aux États-Unis.

2. Inventaire en cycle de vie

Les impacts environnementaux d'une activité humaine sont déterminés en réalisant un inventaire de ses *flux élémentaires* (Guinée, 2002 ; Heijungs, 2002). Un flux élémentaire correspond à une quantité de matière rejetée vers l'environnement, telle qu'une quantité de CO_2 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 $p_0, \ldots, p_n$ les prédécesseurs de p (aussi appelés *processus amonts* de p), $c_0, \ldots, c_n$ les coefficients de dépendance entre les processus amonts et p. On note $F_c(p_0), \ldots, F_c(p_n)$ 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 CO_2 émise et la deuxième à la quantité de CH_4 émise. Ces deux flux élémentaires sont exprimés en kg de gaz émis dans l'atmosphère. On a :

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

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 :

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

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 :

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

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

$$F_c(p_{ec}) = \begin{pmatrix} 1,14 \\ 0.0018 \end{pmatrix} \quad F_c(p_{ep}) = \begin{pmatrix} 0,935 \\ 0.00895 \end{pmatrix} \tag{5}$$

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. Comme expliqué dans (Peters, 2007), la méthode la plus efficace repose sur des algorithmes de résolution itératifs.

La modélisation ACV peut faire apparaître des cycles entre les processus. C'est le cas, par exemple, pour la production d'électricité qui requiert de l'électricité. Une telle

modélisation n'a de sens que si un algorithme de résolution itérative converge sur le système d'équations lui correspondant. Par exemple, si pour produire un kWh d'électricité il était nécessaire de consommer plus d'un kWh d'électricité, un algorithme itératif ne pourrait converger. La condition nécessaire pour assurer cette convergence est que le rayon spectral de la matrice des coefficients de ce système soit inférieur à un (Varga, 2010). D'un point de vue physique, cette condition nous garantit que, dans notre modèle, nous ne consommons pas plus que nous ne produisons.

3. 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, l'ensemble des arcs E_d aux relations de dépendance entre les processus et l'ensemble des pondérations correspond à l'ensemble des coefficients. Soit p et $p_0, \ldots, p_n$ 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. Afin de regrouper sémantiquement les processus, nous avons choisi de les indexer avec un ensemble de mots-clefs qui sont stockés dans une ontologie (Gruber, 1993 ; McGuinness, 2003) (qui correspond au troisième graphe). Le vocabulaire de cette ontologie est composé de mots-clefs et de prédicats pour créer des relations binaires entre ces mots-clefs.

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-clefs. 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.

3.1. Exemple de macro-relation

Reprenons l'exemple des relations de dépendance entre les processus de production d'électricité et les processus de transport présenté dans la section précédente. Au lieu de créer des relations de dépendance entre ces quatre processus individuels, 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 1 présente cette macro-relation.

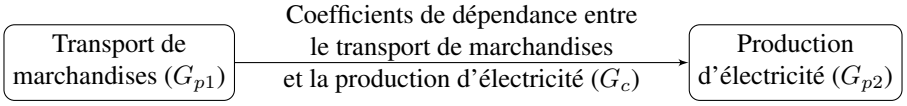

Figure 1. Relation de dépendance entre le groupe des processus de transport de marchandises et le groupe des processus de production d'électricité

Les deux groupes de processus et le groupe de coefficients de cette macro-relation sont représentés dans la figure 2. 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. Les termes que l'on retrouve dans les dimensions de ces tableaux correspondent à des mots-clefs. La notation p_{Train} utilisée dans cette représentation signifie que ce processus, référencé dans G_{p1}, est indexé par le mot-clef *train*. De la même façon, le coefficient $c_{\text{Train,Électricité,Charbon}}$, référencé dans G_c, est indexé par les mots-clefs *Train*, *Électricité* et *Charbon*.

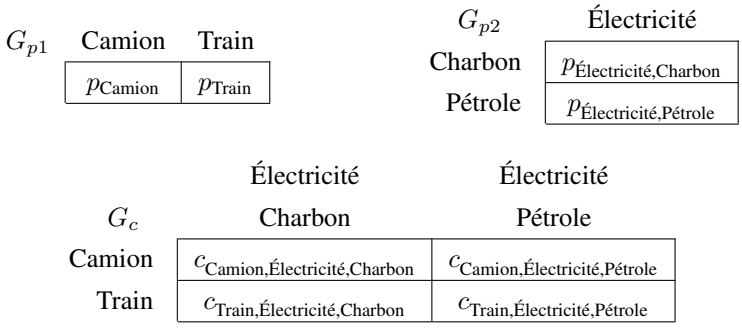

Figure 2. Représentation graphique d'un groupe de coefficients G_c et de deux groupes de processus G_{p1} et G_{p2}

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 3.

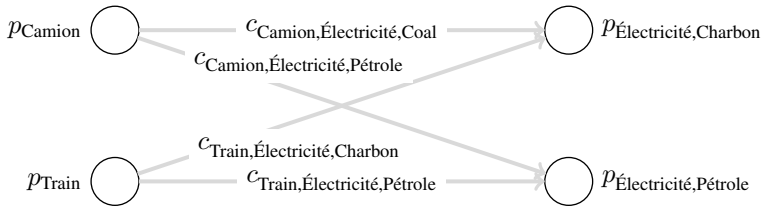

Figure 3. Graphe détaillé de la production d'électricité à partir de pétrole restreint aux processus amonts de transport de marchandises

3.2. Détails de la méthode

La méthode basée sur ces trois graphes est présentée dans la figure 4. Avec cette méthode, la création d'une base d'inventaires repose sur six étapes :

1. créer une ontologie de mots-clefs ;
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 de processus et 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.

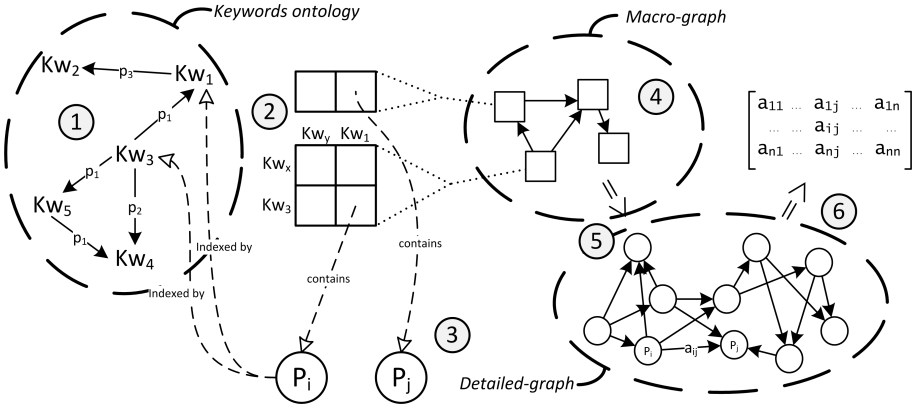

Figure 4. Les six étapes de la méthode proposée

L'existence, éventuelle, de cycles nous empêche de calculer directement les flux cumulés des processus dans le macro-graphe, ce qui nous contraint à le convertir en un graphe détaillé et à calculer les flux cumulés avec une méthode itérative ou directe de résolution d'un système d'équations linéaires.

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 la base de données. Il est aussi possible de réutiliser des groupes déjà définis précédemment pour créer de nouvelles relations de dépendance. Le macro-graphe a également l'avantage de consolider la modélisation. Par exemple, si nous devions modéliser les relations de dépendance entre tous les modes de transport et tous les modes de production d'électricité, le grand nombre de relations entre processus à créer pourrait induire des erreurs (ou des oublis). Alors que 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, donc, réduit le nombre potentiel d'erreurs de modélisation.

4. Formalisme

4.1. Indexation sémantique et ontologie

Dans notre approche, les processus et les coefficients sont indexés et identifiés par des mots-clefs. Nous pouvons donc définir la notion de processus indexé :

DÉFINITION 1. — *Soit $k_1, \ldots, k_n$ des mots-clefs. Un processus indexé est un couple composé de flux élémentaires et d'un ensemble de mots-clefs que l'on note : $p = (F(p), K_p)$ où $K_p = \{k_1, \ldots, k_n\}$.*

Un coefficient n'est associé qu'à une valeur scalaire. De façon analogue à la définition d'un processus indexé, nous pouvons définir un coefficient indexé :

DÉFINITION 2. — *Soit $k_1, \ldots, k_n$ 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 $K_c = \{k_1, \ldots, k_n\}$.*

Par exemple, le processus de transport par camion est identifié par le mots-clef : *Camion*. Il s'écrit donc sous la forme : $p_{\text{Camion}} = (F(p_{\text{Camion}}), \{Camion\})$. 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 précis de mots-clefs. Cette contrainte est nécessaire pour la conversion des macro-relations et le déréférencement des éléments d'un groupe. Donc, il n'y a potentiellement qu'un seul processus ou coefficient correspondant à un ensemble de mots-clefs.

Le vocabulaire de notre ontologie s'exprime aisément en RDF (Klyne, 2004) avec les triplets suivants exprimés dans la syntaxe turtle (Beckett, 2011) [2] :

2. Nous utilisons un espace de nommage XML vide pour présenter les concepts de notre ontologie.

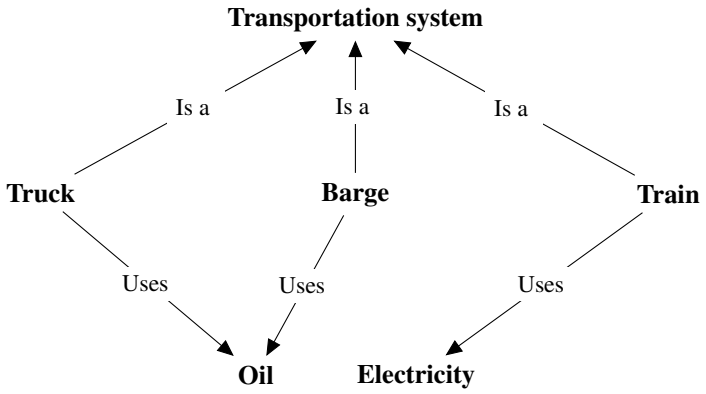

Figure 5. Exemple d'ontologie pour stocker les mots-clefs décrivant des modes de transport. Les nœuds sont des mots-clefs et les étiquettes des arcs sont des prédicats

```
:Keyword rdf:type rdfs:class;
:Predicate [ rdf:type rdf:Property;
             rdfs:range :Keyword;
             rdfs:domain :Keyword ].
```

Nous présentons dans la figure 5 un exemple d'ontologie des mots-clefs servant à décrire différents modes de transport. Les triplets RDF correspondants sont :

```
:modeDeTransport rdf:type :Keyword;
:is_a rdf:type :Predicate;
:camion :is_a :modeDeTransport;
:barge :is_a :modeDeTransport;
:train :is_a :modeDeTransport;
:train :estDeType :terrestre;
:train :estDeType :terrestre;
:train :estDeType :maritime;
```

4.2. Dimensions de mots-clefs

Les dimensions des groupes de processus ou de coefficients, que l'on peut considérer comme des matrices multidimensionnelles, sont des ensembles distincts de mots-clefs. Chaque cellule de ces matrices sera associée à un ensemble de mots-clefs, les coordonnées de la cellule. A chaque coordonnée sera associé un processus ou un coefficient selon que l'on considère un groupe de processus ou un groupe de coefficient.

Plus formellement, soit S une ontologie, on note $\mathscr{P}(S)$ l'ensemble des sous-ensembles de S (privé de l'ensemble vide). On appelle *dimensions* les éléments de $\mathscr{P}(S)$ et on note $\mathscr{P}(\mathscr{P}(S))$ l'ensemble des sous-ensembles de $\mathscr{P}(S)$.

Un ensemble de dimensions n'est valide que s'il a des dimensions distinctes, *i.e.* : si ses dimensions n'ont pas de mots-clefs en commun. Nous définissons la notion de consistance d'un ensemble de dimensions telle que :

DÉFINITION 3. — *Un élément* $\mathscr{D} \in \mathscr{P}(\mathscr{P}(S)), \mathscr{D} = (D_0, \ldots, D_n)$*, est dit consistant si* $\forall (i,j) \in \mathbb{N}^2, i \neq j, D_i \cap D_j = \emptyset$.

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 4. — *Deux éléments* $\mathscr{D}$ *et* $\mathscr{D}'$ *de* $\mathscr{P}(\mathscr{P}(S))$ *sont compatibles si :*

$$\forall D \in \mathscr{D},\ Card\{\ D' \in \mathscr{D}' \mid D \cap D' \neq \emptyset\ \} \leq 1$$

Par exemple, soit $\mathscr{D}$, $\mathscr{D}'$ et $\mathscr{D}''$ trois ensembles de dimensions tels que :

$$\mathscr{D} = \{D_1, D_2\} = \{\ \{a,b\}, \{c,e\}\ \}$$
$$\mathscr{D}' = \{D_3\} = \{\ \{a,f\}\ \} \text{ et } \mathscr{D}'' = \{D_4\} = \{\ \{b,c\}\ \}$$

Les ensembles $\mathscr{D}$ et $\mathscr{D}'$ sont compatibles, alors que $\mathscr{D}$ et $\mathscr{D}''$ sont incompatibles car $D_1 \cap D_4 = \{b\}$ et $D_2 \cap D_4 = \{c\}$.

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 5, nous pouvons créer une dimension contenant tous les modes de transport à l'aide d'une requête récupérant tous les mots-clefs reliés au mot-clef *Mode de transport* en ne considérant que le prédicat *is a*. Nous pouvons aussi ne récupérer que les modes de transport terrestre 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*. Les dimensions peuvent être exprimées *via* des requêtes SPARQL (Prud'hommeaux, Seaborne, 2008) sur l'ontologie exprimée en RDF. Par exemple, la requête permettant de récupérer tous les mots-clefs décrivant des modes de transport qui consomment du pétrole est :

```
SELECT ?keyword
WHERE { ?keyword :is_a :modeDeTransport.
        ?keyword :estDeType :terrestre. }
```

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 5 (tel que *avion*), chaque groupe qui possède une dimension contenant les modes de transport sera mis à jour.

4.3. Groupes de processus et de coefficients

Avec les notions de dimension et de consistance d'un ensemble de dimensions, nous définissons la notion de groupe de processus telle que :

DÉFINITION 5. — *Soit P l'ensemble des processus. On appelle groupe de processus le couple $(\mathscr{D}, P_p)$ où $\mathscr{D} = (D_0, \ldots, D_n)$ est un ensemble consistant de dimensions et P_p est une application $P_p : D_0 \times \cdots \times D_n \to P$.*

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

$$G_p = (\mathscr{D},\ P_p) = (\ \{D_1, \ldots, D_n\},\ \{p_1, \ldots, p_n\}\ )$$
$$G_p = (\ \{D_1, \ldots, D_n\},\ \{\ (F(p_1), K_{p_1}), \ldots, (F(p_n), K_{p_n})\ \}\ )$$

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

DÉFINITION 6. — *Soit C l'ensemble des coefficients. On appelle groupe de coefficients le couple $(\mathscr{D}, C_c)$ où $\mathscr{D} = (D_0, \ldots, D_n)$ est un ensemble consistant de dimensions et C_c est une application $C_c : D_0 \times \cdots \times D_n \to C$.*

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

$$G_c = (\mathscr{D}, C_c) = (\ \{D_1, \ldots, D_n\},\ \{c_1, \ldots, c_n\}\ )$$
$$G_c = (\ \{D_1, \ldots, D_n\},\ \{\ (V(c_1), K_{c_1}), \ldots, (V(c_n), K_{c_n})\ \}\ )$$

Le groupe de processus G_{p1} et le groupe de coefficients G_c que nous avons présenté dans la troisième section s'écrivent alors :

$$G_{p1} = (\mathscr{D}_{p1},\ P_{p1}) = \ (\ \{D_{p1_1}\},\ P_{p1}) = \ (\ \{\ \{\textit{Camion, Train}\}\ \},\ P_{p1}\ )$$
$$G_c = (\mathscr{D}_c,\ C_c) = \ (\ \{D_{c_1},\ D_{c_2},\ D_{c_3}\},\ C_c)$$
$$G_c = (\ \{\ \{\textit{Camion, Train}\},\ \{\textit{Électricité}\},\ \{\textit{Charbon, Pétrole}\}\ \},\ C_c\ )$$

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} n'a une intersection non nulle qu'avec D_{c_1}. 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} n'est valide que 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}).

4.4. 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 contenant des processus correspondant à la phase d'extraction et le deuxième 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 ne comportant qu'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 $\mathscr{C}$ et, dans la notation d'un groupe, nous les faisons apparaître au même titre que les dimensions. Un groupe de processus est donc noté :

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

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

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

5. 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 coefficients [3] 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 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. 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 $a_{ij} = c$.

5.1. Union de deux ensembles de dimensions

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

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.

DÉFINITION 7. — *On note $\cup_D$ l'opérateur d'union de deux ensembles de dimensions tel que, pour deux ensembles de dimensions $\mathscr{D}_1$ et $\mathscr{D}_2$:*

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

Par exemple, soit $\mathscr{D}_1$ et $\mathscr{D}_2$ deux ensembles de dimensions tels que $\mathscr{D}_1 = \{D_1, D_2\}$ $= \{ \{a, b, c\}, \{e, f\} \}$ et $\mathscr{D}_2 = \{D_3, D_4\} = \{ \{a, b, d\}, \{g, h\} \}$. On a : $\mathscr{D}_1 \cup_D \mathscr{D}_2 =$ $\{ \{a, b\}, \{e, f\}, \{g, h\} \}$.

5.2. 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 8. — *On note $\alpha(\mathscr{D}_1, \mathscr{D}_2)$ le nombre d'appariements entre deux ensembles de dimensions $\mathscr{D}_1$ et $\mathscr{D}_2$ tel que :*

$$\alpha(\mathscr{D}_1, \mathscr{D}_2) = \textit{card}(\{ (D_1, D_2) \mid D_1 \in \mathscr{D}_1 \wedge D_2 \in \mathscr{D}_2 \wedge D_1 \cap D_2 \neq \emptyset \})$$

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

5.3. 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 :

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

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.

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 $\alpha(\mathscr{D}_1, \mathscr{D}_c)$, 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 mot-clefs qu'il y a de dimensions dans le groupe. L'ensemble des dimensions d'un groupe étant consistant (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 :

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

6. Algorithme 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'appariement 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 conversion. Pour chacun de ces algorithmes nous prenons comme point de départ une version naïve de leurs implémentations que nous optimisons. Une approche déclarative basée sur l'algèbre relationnelle et une implémentation en SQL de ces algorithmes ont été présentées dans (Bertin, 2012b).

6.1. Calcul du nombre d'appariements

Le calcul du nombre d'appariements entre deux ensembles de dimensions $\mathscr{D}_1$ et $\mathscr{D}_2$ 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 $\mathscr{D}_1$ a une intersection non nulle avec une dimension de $\mathscr{D}_2$ en parcourant tous les mots-clefs de toutes les dimensions de $\mathscr{D}_1$ et de $\mathscr{D}_2$.

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

– en stockant tous les mots-clefs $\mathscr{D}_2$ dans un dictionnaire (ou tableau associatif) [4] et en testant si l'un des mots-clefs de chaque dimensions de $\mathscr{D}_1$ se retrouve dans ce dictionnaire. Les deux ensembles étant compatibles, si on trouve un mot-clef d'une dimension de $\mathscr{D}_2$ dans $\mathscr{D}_1$ 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 $\mathscr{D}_1$ est présent dans l'ensemble de dimensions de $\mathscr{D}_2$, nous pouvons passer à la dimension suivante de $\mathscr{D}_1$.

Dans tous les cas nous devons parcourir toutes les dimensions de $\mathscr{D}_1$ et tous les mots-clefs de ces dimensions pour tester si on a une intersection non nulle avec une dimension de $\mathscr{D}_2$. Dans le pire des cas, qui correspond à un nombre d'appariements égal à zéro [5], il est nécessaire de parcourir tous les mots-clefs des dimensions de $\mathscr{D}_1$. En considérant que la recherche d'un élément dans un dictionnaire se fait dans le pire des cas en $O(\log(n))$ [6], la complexité dans le pire des cas est donc $O(m \log(n))$ avec $n = \sum_{D \in \mathscr{D}_2} Card(D)$ et $m = \sum_{D \in \mathscr{D}_1} Card(D)$.

6.2. Union de deux ensembles de dimensions

De façon similaire, nous pouvons calculer l'union de deux ensembles de dimensions $\mathscr{D}_1$ et $\mathscr{D}_2$ en utilisant un dictionnaire pour déterminer si un mot-clef de $\mathscr{D}_1$ se trouve dans une dimension de $\mathscr{D}_2$. 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 $\mathscr{D}_1$ en construisant l'ensemble de dimensions résultant de cette opération, et :

---

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 n'est en faite utilisée que pour accélérer le processus de recherche d'un mot-clef dans l'ensemble des mots-clefs d'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 $O(n)$.

– si une dimension de $\mathscr{D}_1$ a une intersection non nulle avec une dimension de $\mathscr{D}_2$ (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 $\mathscr{D}_2$ correspondante comme déjà utilisée ;

– si une dimension de $\mathscr{D}_1$ n'a aucune intersection non vide avec une dimension de $\mathscr{D}_2$, nous l'ajoutons au résultat ;

– nous rajoutons toutes les dimensions de $\mathscr{D}_2$ non utilisées dans le résultat.

Dans tous les cas, toutes les dimensions de $\mathscr{D}_1$ et tous leurs mots-clefs seront parcourus. Le pire des cas est trouvé quand l'union de $\mathscr{D}_1$ et $\mathscr{D}_2$ est égale à l'union de toutes leurs dimensions [7]. Il est alors nécessaire de parcourir en plus toutes les dimensions de $\mathscr{D}_2$. La complexité dans le pire des cas est donc $O(n\log(n) + m)$ avec $n = \sum_{D\in\mathscr{D}_1} Card(D)$ et $m = Card(\mathscr{D}_2)$. En considérant qu'on a $m << n$, la complexité dans le pire des cas est $O(n\log(n))$.

6.3. Conversion d'une macro-relation

Une version naïve de l'algorithme de conversion d'une macro-relation $((G_{p1}, G_{p2}), G_c)$ en un ensemble de relations détaillées consiste à parcourir tous les éléments des groupes. Cette version nécessite de tester l'association de $|P_{G_{p1}}| * |P_{G_c}| * |P_{G_{p2}}|$ triplets contenant un processus amont, un coefficient et un processus aval. 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 $O(n^3)$.

Afin d'optimiser cet algorithme, nous stockons les coefficients et les processus avals 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 $\mathscr{D}_{p1}$ et $\mathscr{D}_c$ 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 $\mathscr{D}_{p1} \cup_D \mathscr{D}_c$ et $\mathscr{D}_{p2}$.

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 $\mathscr{D}_{p1} \cup_D \mathscr{D}_c$.

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 lexicographique [8]. Donc, pour un

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

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 totalement dans un ordre lexicographique.

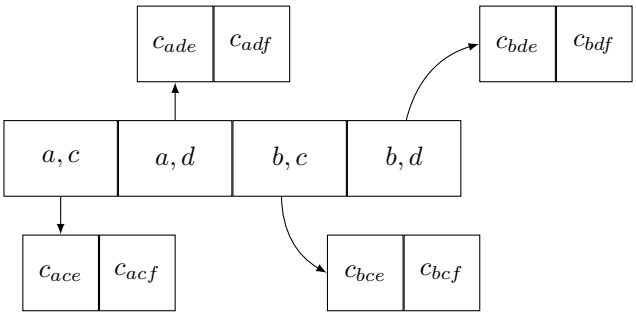

Figure 6. Dictionnaire des coefficients indexés par les mots-clefs communs avec le groupe de processus amont

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 avec le résultat de la multiplication d'un processus amont par 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 telle que :

$$G_{p1} = (\mathscr{D}_{p1},\ P_{p1})$$
$$G_c = (\mathscr{D}_c,\ C_c)$$

avec :

$$\mathscr{D}_{p1} = \{D_{p1_1}, D_{p1_2}\} = \{\{a, b\}, \{c, d\}\}$$
$$P_{p1} = \{p_{ac}, p_{ad}, p_{bc}, p_{bd}\}$$
$$\mathscr{D}_c = \{D_{c_1}, D_{c_2}\} = \{\{a, b\}, \{c, d\}, \{e, f\}\}$$
$$C_c = \{c_{ace}, c_{acf}, c_{ade}, c_{adf}, c_{bce}, c_{bcf}, c_{bde}, c_{bdf}\}$$

L'ensemble des mots-clefs communs entre ces deux groupes est $\{a, b, c, d\}$. Les coefficients de G_c sont stockés dans un tableau associatif, la figure 6 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_{p_1} et un groupe de coefficients G_c tels que :

$$G_{p1} = (\mathscr{D}_{p1},\ P_{p1}) = (\ \{\{a,b\},\{c,d\}\},\ P_{p1}\ )$$
$$G_c = (\mathscr{D}_c,\ C_c) = (\ \{\{a,b\},\{c,e\}\},\ C_c\ )$$

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, nous devons donc déterminer si le nombre de mots-clefs qui appartiennent à cette clef est égal au nombre d'appariements.

Cette optimisation peut aussi être appliquée à la recherche de processus avals 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 avals en fonction de ces mots-clefs communs et du nombre d'appariements $\alpha(\mathscr{D}_{p1} \cup_D \mathscr{D}_c, \mathscr{D}_{p2})$. Ici aussi, nous réduisons l'espace de recherche aux seuls processus avals éligibles pour une relation détaillée étant donné un processus amont et un coefficient.

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 $\alpha(\mathscr{D}_{G_{p1}}, \mathscr{D}_{G_c}) = 0$;

– il n'y a aucun mot-clef commun entre le résultat de la multiplication G_{p_1} par G_c et le groupe de processus aval. Dans ce cas nous avons $\alpha(\mathscr{D}_{G_{p1}} \cup_D \mathscr{D}_{G_c}, \mathscr{D}_{G_{p2}}) = 0$;

– la combinaison des deux cas précédents. Dans ce cas nous avons $\alpha(\mathscr{D}_{G_{p1}}, \mathscr{D}_{G_c}) = 0$ et $\alpha(\mathscr{D}_{G_{p1}} \cup_D \mathscr{D}_{G_c}, \mathscr{D}_{G_{p2}}) = 0$.

Dans ces cas-là, la version naïve de l'algorithme crée des relations détaillées pour chaque combinaison d'éléments. L'utilité d'une macro-relation correspondant à l'un de ces cas peut sembler assez limitée. Mais pour assurer que l'optimisation proposée ait le même comportement que la version naïve de l'algorithme, nous devons rajouter un traitement spécifique.

Dans le cas où 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. Nous présentons ci-après l'algorithme de conversion dans sa version optimisée.

Algorithme 1: Conversion d'une macro-relation en un ensemble de relations détaillées

Entrées : G_{P1},G_{P2},G_c
relationsDétaillées ← ∅
dictCoeffs ← Dictionnaire des coefficients
dictProcessusAval ← Dictionnaire des processus avals
pour chaque $p_1 \in G_{P1}$.processus **faire**
  clefHachageCoeff ← Clef de hachage de p_1
  si clefHachageCoeff *not null* **alors**
    pour chaque c dictCoeffs [clefHachageCoeff] **faire**
      clefHachageProcessusAval ← Clef de hachage de p_1
      si clefHachageProcessusAval *not null* **alors**
        pour chaque $p_2 \in$ dictProcessusAval [clefHachageProcessusAval] **faire**
          relationsDétaillées += ((p_1,p_2),c)
retourner relationsDétaillées

Si les groupes n'ont aucun mots-clefs 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. Dans ce contexte, nous estimons que la complexité moyenne est inférieure à $O(n^3)$.

7. 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 sur des jeux de données synthétiques pour 1) vérifier l'impact des optimisations apportées à la version naïve ; 2) nous assurer que nous pouvons convertir des macro-relations de tailles variables dans un temps raisonnable.

Au préalable, quelques remarques s'appliquent à ces deux 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 (Frischknecht, 2005b) ;

– 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-clefs, l'objectif principal étant de tester le passage à l'échelle sur le traitement des coordonnées des éléments à associer dans des relations détaillées ;

– l'expérimentation a été réalisée en PHP sous Linux sur un PC équipé d'un Intel Xéon E31245 à 3.30GHz avec 8Go de mémoire vive.

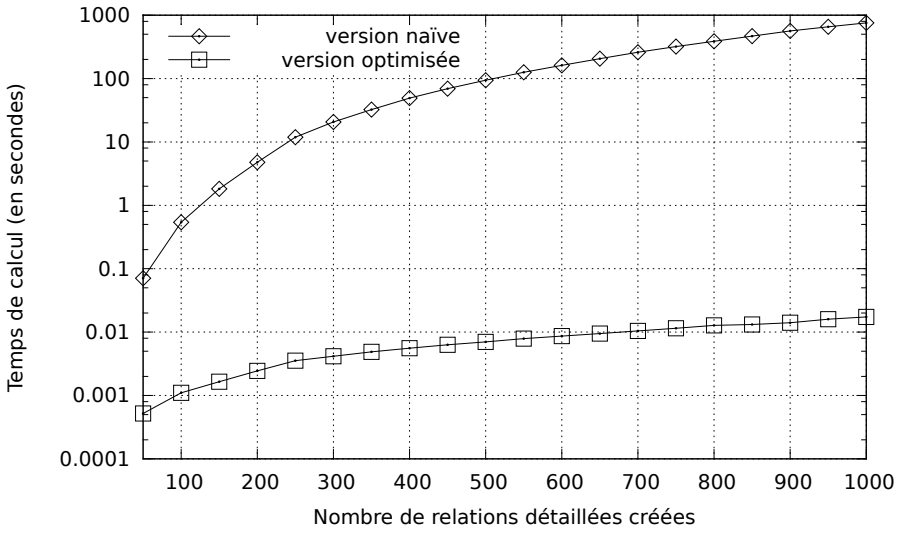

Figure 7. Comparaison du temps de calcul pour convertir des macro-relations de taille croissante entre la version naïve et la version optimisée de l'algorithme de conversion des macro-relations. L'échelle de l'axe des ordonnées est logarithmique

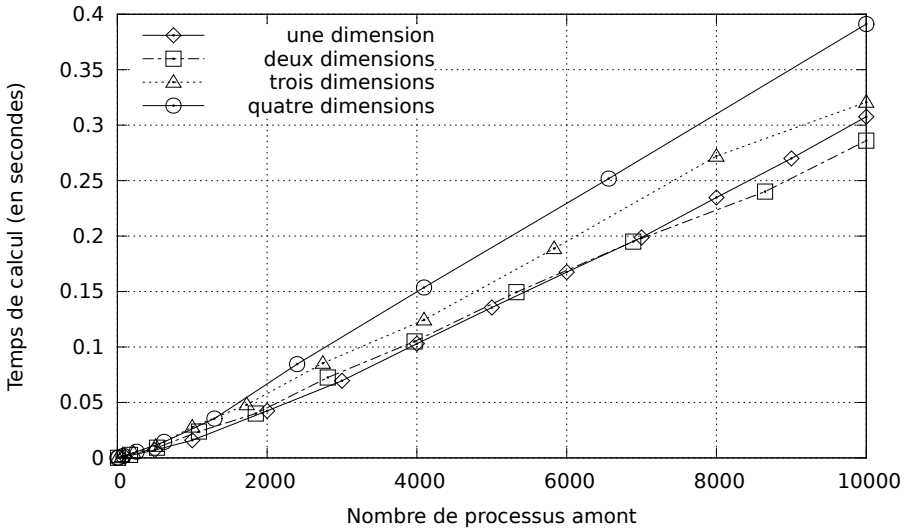

Figure 8. Comparaison du temps de calcul pour convertir des macro-relations de taille croissante avec la version optimisée de l'algorithme de conversion pour des relations impliquant des groupes de une, deux, trois et quatre dimensions

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 impliquant tous les éléments des groupes : 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.

Dans le graphique présenté figure 7, 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 1 000 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 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 8. 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.

8. Etude de cas : production d'électricité aux Etats-Unis

Nous avons étudié l'application de l'approche multiniveaux 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). Cette base contient des processus pour un ensemble d'activités économiques aux États-Unis. Elle contient environ 600 processus accessibles librement depuis une interface web. Les données d'inventaires pour chaque processus peuvent être exportées dans des fichiers Excel ou XML respectant le format ecospold (Frischknecht, 2005a). 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.

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 (NREL, 2011). Ces sous-regions sont définies par l'agence de protection de l'environnement américaine et la Generation Resource Integrated Database (eGRID). Chaque sous-région de l'eGRID représente des

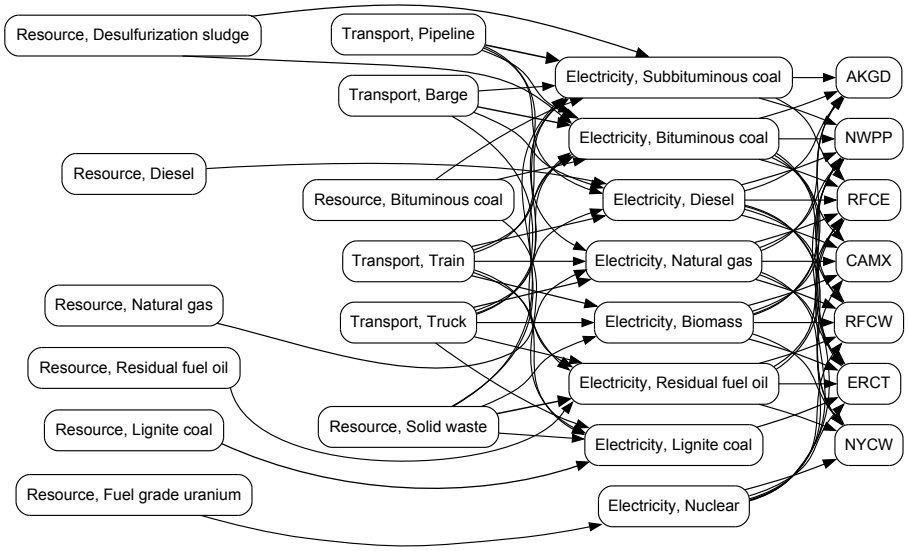

Figure 9. Graphe détaillé correspondant au jeu de données extrait de la base de données d'inventaires en cycle de vie du NREL

portions du territoire américain qui ont des mix énergétiques similaires et qui sont potentiellement isolées en raison de contraintes de transmission.

Certains processus de cette base de données ne sont pas détaillés (et marqués comme "factice"), 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 9 présente le graphe détaillé correspondant à ce jeu de données limité 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.

Nous proposons de créer un macro-graphe qui mette en avant les grandes familles de ressources utilisées pour produire de l'électricité. Le macro-graphe que nous obtenons est présenté dans la figure 10. Ce 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.

8.1. 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

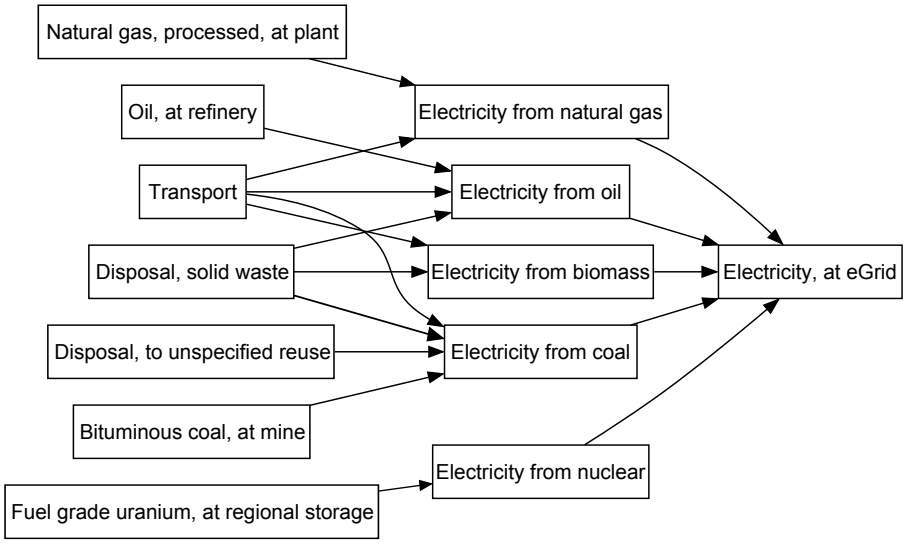

Figure 10. Macro-graphe correspondant au jeu de données extrait de la base de données d'inventaires en cycle de vie du NREL

processus G_{p1} 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. Le groupe de processus G_{p2} 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.

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 11. 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 valeur 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.

8.2. 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 12 présente les groupes impliqués dans cette macro-relation et les relations détaillées obtenues

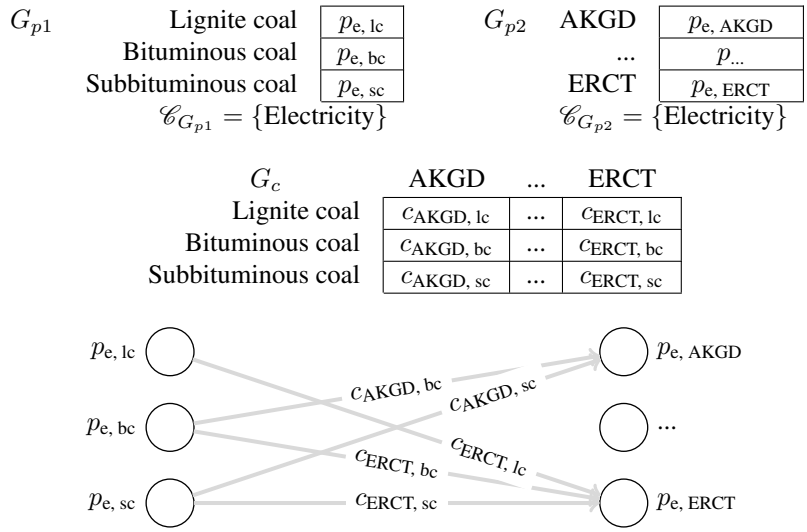

Figure 11. Conversion en relations détaillées d'une macro-relation $((G_{p1}, G_{p2}), G_c)$. *G_{p1} contient les processus de production d'électricité à partir de charbon. G_{p2} contient les processus de production d'électricité pour les sous-régions de l'eGrid. G_c contient les coefficients de dépendance 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 simplifié : $p_{e,lc}$ correspond au processus $p_{\text{Electricity, Lignite coal}}$*

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 transport. 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.

8.3. Macro-modélisations alternatives

Il est aussi possible de réduire le nombre de groupes de coefficients nécessaire à 12. Par exemple, le groupe de processus des 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 1). Nous pouvons aussi utiliser seulement 8 groupes de coefficients si nous stockons dans un seul groupe tous

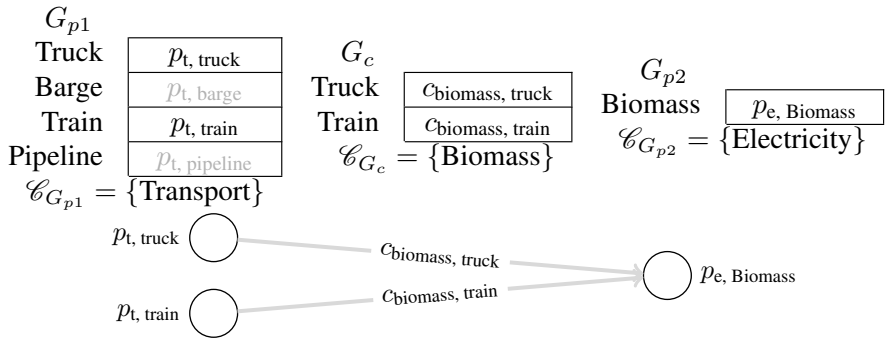

Figure 12. Conversion en relations détaillées d'une macro-relation entre les processus de transport de marchandises et le processus de production d'électricité à partir de biomasse

les coefficients de dépendance 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 n'existe pas un unique macro-graphe correspondant à ce jeu de données. Nous pouvons, par exemple, créer un macro-graphe dans lequel nous classons les moyens de production d'électricité renouvelables dans un groupe et les non renouvelables dans un autre. Il est aussi possible de regrouper les sous-régions de l'eGrid en fonction de leur emplacement géographique.

9. Conclusion et travaux futurs

Nous avons proposé une nouvelle approche pour modéliser les inventaires en cycle de vie basée sur l'utilisation d'une ontologie et des regroupements sémantiques des processus. L'intérêt principal de cette proposition est d'offrir un modèle plus compréhensible des bases de données d'inventaires. Cette approche permet aussi d'avoir une modélisation basée sur la manipulation d'une ontologie pour créer de nouvelles relations entre des processus. Ceci nous permet de réduire le nombre de relations de dépendance que doit créer et maintenir un éditeur de la base, mais l'utilisation d'une ontologie complexifie son travail.

Dans un futur proche nous souhaitons pouvoir exécuter des requêtes sémantiques sur le graphe détaillé pour tirer parti de l'indexation sémantique des processus. Plus précisément, nous souhaiterions savoir quel est l'impact des processus indexés avec un mot-clef (ou un ensemble de mots-clefs) sur un processus spécifique. Nous pourrions, par exemple, déterminer quel est l'impact des processus de transport sur la production d'électricité pour une sous-région spécifique de l'eGRID. A plus long terme, nous envisageons de générer un macro-graphe automatiquement, ou semi-automatiquement, à partir d'une base d'inventaires déjà existante et d'ontologies déjà existantes.

Bibliographie

Beckett D. (2011). *Turtle terse rdf triple language*. W3C Recommendation. World Wide Web Consortium. Consulté sur http://www.w3.org/TR/rdf-sparql-query/

Bertin B., Scuturici V.-M., Pinon J.-M., Risler E. (2012a). Carbondb: a semantic life cycle inventory database. In *CIKM*, p. 2683–2685.

Bertin B., Scuturici V.-M., Risler E., Pinon J.-M. (2012b). A semantic approach to life cycle assessment applied on energy environmental impact data management. In *EDBT/ICDT Workshops*, p. 87–94.

Frischknecht (2005a). Introduction the ecoinvent database: Overview and methodological framework. *The International Journal of Life Cycle Assessment*, vol. 10, n° 1, p. 3–9.

Frischknecht R., Rebitzer G. (2005b). The ecoinvent database system: a comprehensive web-based lca database. *Journal of Cleaner Production 2005*, vol. 13, n° 13, p. 1337–1343.

GaBi. (2011). *Gabi life cycle inventory databases.* Consulté sur http://www.gabi-software.com

Gruber T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, vol. 5, n° 2, p. 199–220.

Guinée J. B. (2002). *Handbook on life cycle assessment: Operational guide to the iso standards*, New York, Springer, p. 708.

Heijungs R., Suh S. (2002). *The computational structure of life cycle assessment*, Dordrecht, Kluwer Academic Publishers, p. 256.

ISO. (2006). *Iso 14044 (2006): Environmental management life cycle assessment requirements and guidelines*. International Standard. International Organisation for Standardisation.

Klyne G., Carroll J. J. (2004). *Resource description framework (rdf): Concepts and abstract syntax*. W3C Recommendation. World Wide Web Consortium. Consulté sur http://www.w3.org/TR/rdf-concepts/

Leontief W. (1986). Input-output analysis. In *Regional economic impact analysis and project evaluation*, p. 53–83. University of British Columbia Press.

McGuinness D. L. (2003). Ontologies come of age. In H. L. W. W. D. Fensel J.A. Hendler (Ed.), *Spinning the semantic web: Bringing the world wide web to its full potential*, MIT Press, p. 171–194.

Nicholson W. (1990). *Elementary linear algebra with applications*, PWS-Kent Publishing Company, p. 576.

NREL. (2011). *U.s. life cycle inventory database.* Consulté sur http://www.nrel.gov/lci/

Peters G. P. (2007). Efficient algorithms for life cycle assessment, input-output analysis, and monte-carlo analysis. *The International Journal of Life Cycle Assessment*, vol. 12, n° 6, p. 373–380.

Prud'hommeaux E., Seaborne A. (2008). *Sparql query language for rdf*. W3C Recommendation. World Wide Web Consortium. Consulté sur http://www.w3.org/TR/rdf-sparql-query/

Varga R. (2010). *Matrix iterative analysis*, Springer, p. 358.