A Semantic Approach to Life Cycle Assessment Applied on Energy Environmental Impact Data Management

Benjamin Bertin
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205,
F-69621, France
benjamin.bertin
@liris.cnrs.fr

Vasile-Marian Scuturici
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205,
F-69621, France
marian.scuturici
@liris.cnrs.fr

Jean-Marie Pinon
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205,
F-69621, France
jean-marie.pinon
@liris.cnrs.fr

Emmanuel Risler
Université de Lyon, CNRS
INSA-Lyon, ICJ, UMR5208,
My C-Sense, F-69621, France
emmanuel.risler@insa-lyon.fr

ABSTRACT

Environmental impact assessment of goods and services is nowadays a major challenge for both economic and ethical reasons. Life Cycle Assessment (LCA) provides a well-accepted methodology for modelling environmental impacts of human activities. One stage of the LCA methodology is Life Cycle Inventory, which consists of decomposing economic activities as elementary processes linked together through interdependency relations. This stage is needed in order to evaluate the environmental impacts of processes. A global analysis of economic activities requires considering a huge amount of elementary processes and interdependency links, making the model difficult to understand. In this paper, we propose a semantic approach for the modelling of lice cycle inventory databases. The method has the advantage of offering a more comprehensible model. We explain our model and illustrate it with life cycle inventory data for the U.S. electricity production.

Keywords

Energy impact data management, Life Cycle Assessment, Ontology

1. INTRODUCTION

In order to reduce the environmental impact of human activities, such as greenhouse gases emissions, it is necessary to model and evaluate the environmental effects of those activities. This is the objective of the Life Cycle Assessment (LCA) method[3], which aims at determining the environmental impact of a product, a service or, generally speaking, any human activity. This method can take into account all the life cycle stages of a product such as manufacture, use and recycling. LCA can assess various environmental impacts such as greenhouse gases emissions or chemical products dissemination.

A LCA study is composed of four phases[3]. The first one consists of the definition of the goal and scope of the study. In the second one the studied system is factorized into interrelated elementary processes associated to environmental impacts. This phase is called Life Cycle Inventory (LCI). Elementary processes are related to specific stages of a life cycle or to any human activities (e.g.: energy production, fertilizer spreading, air plane trip, etc.). Processes can be composed of other processes, for instance: car production depends on steel production. The analysis and the interpretation of the LCI data are done in the third and the fourth phases.

The accepted methodology for the LCI is to use an Input/Output (I/O) matrix[11] to model inter-processes interactions or interactions between processes and the environment. The first section of this paper contains a brief description of this model.

Several agencies and companies provide basic Life Cycle Inventories for some common processes[6][1][2]. But those databases can contain thousands of processes linked together. The model is therefore difficult to understand unless we do an in-depth analysis. Semantic similarities are noticeable in LCI databases, e.g.: every electricity production from coal (such as the electricity production from bituminous coal or subbituminous coal) processes depends on transportation processes. But it is hard to understand the dependencies between electricity production from coal and transportation processes if this information is scattered into an I/O matrix.

Thus, we propose to add semantic information into LCI model and to semantically regroup processes in order to offer a bird's-eye view of the processes and, as a side effect, a new way to express processes dependencies. Our

EDBT '12, March 27–30 2012, Berlin, Germany.
Copyright 2012 ACM 978-1-4503-1143-4/12/03 ...$10.00

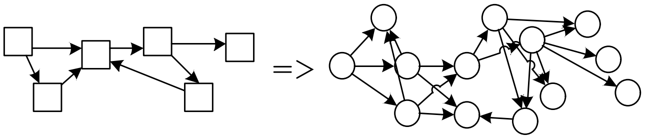

Figure 1: The two digraphs of our approach. The graph on the left side contains groups of processes and relations between these groups, this is the macro-graph. It is translated into the graph on the right side containing processes and inter-processes relations, this is the detailed-graph.

model is based on the coexistence of two digraphs. The first one, called the *detailed-graph*, is a transposition of the Input/Output matrix. The second one, called the *macro-graph*, contains dependencies between regrouped processes. The macro-graph contains macro dependency relations that need to be translated into the detailed-graph in order to calculate the impacts of processes. Figure 1 illustrate the two digraphs layers. The second section explains our model. The third section contains the translation method of the macro-relations contained in the macro-graph into relations between processes in the detailed-graph and offers some key parts of our implementation.

In the last section, we present an application of our approach to electricity production processes in the U.S., extracted from the National Renewable Energy Laboratory LCI Database[2]. This data set contains elementary processes to be used in Life Cycle Assessment in the U.S. and contains inter-processes dependencies. However, we restricted our study to electricity production processes only.

2. LIFE CYCLE INVENTORY MODEL

The goal of Life cycle assessment methodology is to evaluate the environmental impacts of, for instance, a product or a company's activity. This can be achieved by creating an inventory of elementary flows from and to the environment, for every step of a product's production process or for every activity of a company[8][9]. In the LCA terminology, these steps or activity are called *processes.*

Life cycle assessment processes have dependency relations connecting each other, e.g.: it is necessary to extract uranium in order to run a nuclear power plant. The usual model is based on an Input/Output matrix A. For n processes, this is a $n \times n$ matrix, where a_{ij} is the dependency scalar value, called *dependency factor*, between process i and process j. This matrix depicts flows to and from the technosphere[1], these flows are named elementary flows.

These processes are associated with environmental impacts, e.g.: the greenhouse gases emitted or the resulting water pollution. The m different impacts produced by n processes are modelled in a $n \times m$ matrix B where b_{ij} is the j^{th} impact of the i^{th} process. This matrix depicts flows from the technosphere to the ecosphere[2].

Determining the impacts for a specific process requires to recursively calculate the impacts of its predecessor. As explained in [14], the I/O matrix can be considered as a basic system of linear equations. Thus it is possible to calculate the impacts of the processes using iterative methods or any direct method (like the Gaussian elimination)[18].

Interestingly for our proposition, there is one condition for the matrix to be computable: the linear equation system must converge. For instance, if we say that in order to produce one gallon of oil, we need to consume more than one gallon of oil, an iterative algorithm to solve the linear equations system would not converge. The system converges when the spectral radius of the matrix A is less than one[17].

3. THE TWO LAYERS MODEL

The second layer of our model has the goal to offer a simplified view of the detailed-graph. In our approach every process is described by a set of keywords and those keywords are stored in an ontology. The vocabulary of this ontology is composed of keywords and predicates to create binary relations between those keywords.

Using this ontology, we can regroup processes and dependency factors (also called coefficients) into semantic groups. A group of processes is similar to a multi dimensional matrix, where each dimension is a set of keywords described using a query over the ontology. We can create dependencies between those group of process in the same way that we have dependencies between processes. This section covers the basic notions of our approach.

3.1 The detailed graph layer

A process is associated with environmental impacts and with other processes. The total amount of environmental impacts of a process is the sum of the impacts of its predecessors multiplied by a scalar dependency coefficient. The environmental impacts of a process can be expressed as a linear combination of other processes environmental impacts.

Let P be a process and $I(P)$ its cumulated environmental impacts, we denote by $P_0, \ldots, P_n$ the preceding processes of P (otherwise called 'upstream' processes of P), $C_0, \ldots, C_n$ the dependencies coefficients between upstream processes. We denote by $I(P_0), \ldots, I(P_n)$ the cumulated environmental impacts of P's upstream processes, and $I_{\text{direct}}(P)$ the direct impacts of P (*i.e.*: the impacts which are not from its upstream processes). We have:

$$I(P) = I_{\text{direct}}(P) + \sum_{i=0}^{n}(I(P_i) * C_i) \tag{1}$$

Let $G(V, E)$ be the digraph representing composition relations, where the vertices set V is the set of processes, the edges set E is the relations between processes and the set of weights associated to the edges is the set of coefficients. Let P and $P_0, \ldots, P_n$ be vertices, an edge between P_i and P means that the process P depends on P_i (see Figure 2).

For instance, if we focus on the transportation system processes and the electricity production from oil process. Be-

[1] The McGraw-Hill Dictionary of Environmental Science defines this term as: The part of the physical environment affected through building or modification by humans.

[2] The U.S. Environmental Agency defines this term as: The "bio-bubble" that contains life on earth, in surface waters, and in the air.

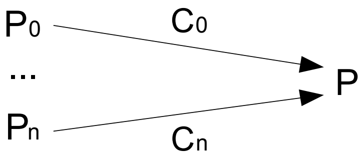

Figure 2: Detailed-graph for process P with $P_0, \ldots, P_n$ as upstream processes and $C_0, \ldots, C_n$ as dependencies coefficients.

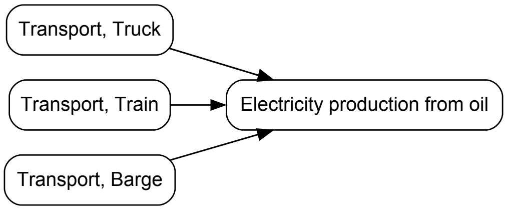

Figure 3: Detailed-graph for the electricity production from oil process restricted to the transportation system processes. Edges are weighted with the dependency coefficients.

cause we have to transport oil from the refinery to the power plant, we have an edge from every transportation system process used for this operation directed to the *electricity production from oil* process (see Figure 3).

3.2 The keywords ontology

In our approach, processes and coefficients are indexed and identified using keywords: there is only one process or coefficient associated to a specific set of keywords, e.g.: the process corresponding to using a truck for 1 mile is indexed using the keywords *Transport* and *Truck*. A group of processes is a multidimensional matrix where dimensions are distinct sets of keywords. Thus, there is only one process or coefficient associated to each coordinate. Keywords are stored in an ontology[7][13] where every keyword can be associated with another one using a predicate.

This keywords ontology is used to dynamically define groups. A set of keywords (called *dimension* here) is the result of a query over the ontology. Considering the ontology in Figure 4, we can build a dimension containing all transportation systems with a query to retrieve all the keywords linked to the keyword *Transportation system* considering only the predicate *is a*. We can express this query in a more compact form: ?*Transportation system.(is a)*. If we want to create a dimension containing only transportation systems running on oil, the query would be: ?*Transportation system.(is a)* ∩ *Oil.(uses)*.

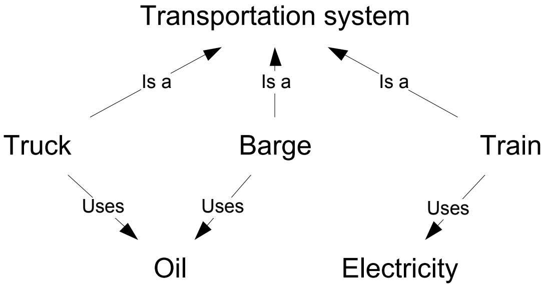

Figure 4: Ontology example for transportation system. Nodes are keywords and edges' labels are predicates.

The vocabulary of our ontology is naturally represented using RDF[10], with the following triples expressed in the turtle syntax[4], assuming that we have an xml namespace *ex* for our ontology:

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

The ontology shown in Figure 4 corresponds to the following RDF statements:

```
ex:transportationSystem rdf:type ex:Keyword;
ex:is_a rdf:type ex:Predicate;
ex:truck ex:is_a ex:transportationSystem;
ex:barge ex:is_a ex:transportationSystem;
ex:train ex:is_a ex:transportationSystem;
ex:train ex:uses ex:oil;
ex:barge ex:uses ex:oil;
ex:train ex:uses ex:electricity;
```

Therefore, the dimensions description is easily done using SPARQL[15]. For instance, the query to get all the transportation systems is:

```
SELECT ?keyword
WHERE { ?keyword es:is_a ex:transportationSystem. }
```

Any change made to the keywords ontology can trigger an update to previously defined dimensions, thus it will update already defined groups. For instance: if a new transportation system is added to the ontology shown in Figure 4 (let's say *air-plane*), every group having a dimension containing transportation system keywords would be updated.

3.3 The macro-graph layer

As previously stated, we can create dependency relations between groups of processes using groups of coefficients. The groups and their dependencies are represented in a weighted digraph $G_M(V, E)$, where the vertices set V is the set containing the groups of processes, the edges set E is the set of relations between groups of processes, and the set of weights associated to the edges is the set of groups of coefficients.

One can easily see the correspondence between the Input/ Output methodology and this approach: the I/O matrix is decomposed into smaller parts which are used as weights in the macro-graph. The macro-graph eases the model comprehension and can be used to model the interactions between processes in order to calculate their environmental impacts.

Because the model can contain cycles between groups, it is not possible to calculate impacts of processes directly on the macro-graph. We need to translate the relations between groups into relations between processes. In other words, we need to translate the edges of the macro-graph into edges in the detailed-graph. Then, we can extract the matrix A in order to compute impacts as it is usually performed in the LCA methodology.

When we create a dependency relation between two groups of processes, we use the set of keywords of the dimensions to know which process of the two groups is to be linked in the detailed-graph as we will explain in the next section.

Group formalism

Let S be an ontology. We denote by $\mathscr{P}(S)$ the set of subsets of S (without the empty set). We call **dimensions** the elements of $\mathscr{P}(S)$ and we denote by $\mathscr{P}(\mathscr{P}(S))$ the set of subsets of $\mathscr{P}(S)$.

A group is valid only if it has distinct dimensions, *i.e.*: if its dimensions have no keywords in common. We define the concept of dimension set consistency as follows:

Definition 1. An element $\mathscr{D} \in \mathscr{P}(\mathscr{P}(S)), \mathscr{D} = (D_0, \dots, D_n)$, is called *consistent* if $\forall (i,j) \in \mathbb{N}^2, i \neq j, D_i \cap D_j = \emptyset$.

Environmental impacts calculation of processes contained in a group of processes having a dependency relation with another group is feasible only if every dimension of both groups has only one match in the other group. A dimension can match another dimension if their intersection is not null. We define the concept of dimensions sets compatibility as follows:

Definition 2. Two elements $\mathscr{D}$ and $\mathscr{D}'$ of $\mathscr{P}(\mathscr{P}(S))$ are *compatible* if the following properties are true:

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

Example: considering $\mathscr{D} = \{D_1, D_2\}$, $\mathscr{D}' = \{D_3\}$, $\mathscr{D}'' = \{D_4\}$, $D_1 = \{A, B\}$, $D_2 = \{C, E\}$, $D_3 = \{A, F\}$ and $D_4 = \{B, C\}$. We say that $\mathscr{D}$ and $\mathscr{D}'$ are compatible, while $\mathscr{D}$ and $\mathscr{D}''$ are incompatible because $D_1 \cap D_4 = \{B\}$ and $D_2 \cap D_4 = \{C\}$.

Having properly defined dimensions and restrictions to the notion of dimension, we can now define what is a group of processes as follows:

Definition 3. Let P be the set of processes. We call *group of processes* an ordered pair $(\mathscr{D}, p)$ where $\mathscr{D} = (D_0, \dots, D_n)$ is a consistent set of dimensions, and p is an application $p : D_0 \times \dots \times D_n \to P$.

Similarly we can define what is a group of coefficients as follows:

Definition 4. Let C be the set of coefficients. We call *group of coefficients* an ordered pair $(\mathscr{D}, c)$ where $\mathscr{D} = (D_0, \dots, D_n)$ is a consistent set of dimensions, and c is an application $c : D_0 \times \dots \times D_n \to C$.

G_{P1}	
Truck	p_{Truck}
Barge	p_{Barge}

G_{P2}	Electricity
Coal	$p_{\text{Electricity,Coal}}$

	Electricity	Electricity
G_C	Coal	Oil
Truck	$c_{\text{Truck,Electricity,Coal}}$	$c_{\text{Truck,Electricity,Oil}}$
Barge	$c_{\text{Barge,Electricity,Coal}}$	$c_{\text{Barge,Electricity,Oil}}$

Figure 5: Graphical representations of a group of coefficient G_C and two groups of processes G_{P1} and G_{P2}. G_{P1} contains transportation processes and is composed of one dimension. G_{P2} contains the electricity production from coal process and is composed of two dimensions. G_C contains coefficients between G_{P1} and the electricity production processes (restricted to the electricity production from coal and from oil) and is composed of two dimensions.

By extending the definition on dimensions set compatibility, we can assume that two groups $G_1 = (\mathscr{D}_1, e_1)$ and $G_2 = (\mathscr{D}_2, e_2)$ are compatible only if $\mathscr{D}_1$ and $\mathscr{D}_2$ are compatible.

Example: Let G_{P1} be a group of processes containing transportation system processes such that $G_{P1} = (\{\{\mathit{Truck,Barge}\}\}, p)$. Let G_C be a group of coefficients containing dependency coefficients between transportation processes and some electricity production processes such that $G_C = (\{\{\mathit{Truck,Barge}\}, \{\mathit{Electricity}\}, \{\mathit{Coal,Oil}\}\}, c)$. We can state that G_{P1} and G_C are compatible. Figure 5 shows a simplified graphical representation of those groups. The notation used in this representation states that the process p_{Truck} referenced in G_{P1} is indexed by the keyword *Truck* and the coefficient $c_{\mathit{Truck,Electricity,Coal}}$ referenced in G_C is indexed by the keywords *Truck*, *Electricity* and *Coal*.

4. TRANSLATION FROM THE MACRO-GRAPH TO THE DETAILED-GRAPH

In LCA theory, a process is associated with a set of environmental impacts (furthermore, each impact is associated with an uncertainty). To explain the translation method in a simpler way, we consider that processes impacts and coefficients are scalar values. The groups definitions are therefore transformed into a simpler form: we call *group* (of processes or coefficients) an ordered pair $(\mathscr{D}, V)$ where $\mathscr{D} = (D_0, \dots, D_n)$ is a consistent set of dimensions, and V is an application $V : D_0 \times \dots \times D_n \to \mathbb{R}$.

For the translation procedure, we use relational algebra to represent groups. A group is stored in one relation with one attribute per dimension and one attribute for the environmental impacts of the processes if it is a group of processes, or one attribute for the coefficients values if it is a group of coefficients. The dimensions attributes form the primary key for these relations.

We also need two relations to store the dimensions keywords set and groups/dimensions associations: a) *Dimensions(Dim, Keyword)* describes dimensions domain of definition, the attribute *Dim* is the primary key; and b) *Groups-*

G_{P1}

$D_{P1}1$	V
Truck	VP_{Truck}
Barge	VP_{Barge}

G_{P2}

$D_{P2}1$	$D_{P2}2$	V
Coal	Electricity	$VP_{\text{Coal,Electricity}}$

G_C

D_C1	D_C2	D_C3	V
Truck	Electricity	Coal	$VC_{\text{Truck,Electricity,Coal}}$
Truck	Electricity	Oil	$VC_{\text{Truck,Electricity,Oil}}$
Barge	Electricity	Coal	$VC_{\text{Barge,Electricity,Coal}}$
Barge	Electricity	Oil	$VC_{\text{Barge,Electricity,Oil}}$

Dimensions

Dim	Keyword
$D_{P1}1$	Truck
$D_{P1}1$	Barge
$D_{P2}1$	Coal
$D_{P2}2$	Electricity
D_C1	Truck
D_C1	Barge
D_C2	Electricity
D_C3	Coal
D_C3	Oil

GroupsDimensions

Group	Dim
G_{P1}	$D_{P1}1$
G_{P2}	$D_{P2}1$
G_{P2}	$D_{P2}2$
G_C	D_C1
G_C	D_C2
G_C	D_C3

Figure 6: Relational representation of a groups G_{P1}, G_{P2} and G_C. VP denotes processes value and VC denotes coefficients value

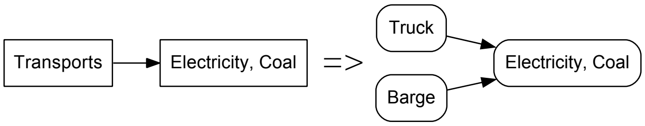

Figure 7: The macro-graph on the left side contains a macro-relation between G_{P1} (noted 'Transports') and G_{P2} (noted 'Electricity, Coal') and is translated into the detailed-graph on the right side (the coefficients are not represented here).

Dimensions(Group, Dim) describes which dimensions are used in groups, the couple $(Group, Dim)$ is the primary key.

Figure 6 shows an example of the relations used to store the groups of processes G_{P1} and G_{P2}, and the group of coefficients G_C shown in Figure 5.

The procedure explained here directly calculates the processes impacts as if an impact was only one scalar value. Thus we are not dealing with cycles. In order to get the detailed-graph's relations we can replace the scalar values by *pointers* to processes or coefficients, the implementation gets a bit more complicated, but the core of the method is the same.

Considering a macro-relation between two groups of processes (an upstream group and a destination group), the calculation procedure is split in two steps: 1) the upstream group is multiplied by the group of coefficients; this gives us a group of processes G_P; and 2) G_P is projected into the destination group.

For instance, a macro-relation between the group G_{P1} and the group G_{P2} found in Figure 5 is weighted with the group of coefficients G_C. Figure 7 shows this relation and the expected relations in the detailed-graph.

	Electricity	Electricity
G_P	Coal	Oil
Truck	$VP_{Truck}*$ $VC_{\text{Truck,Electricity,Coal}}$	$VP_{Truck}*$ $VC_{\text{Truck,Electricity,Oil}}$
Barge	$VP_{Barge}*$ $VC_{\text{Barge,Electricity,Coal}}$	$VP_{Barge}*$ $VC_{\text{Barge,Electricity,Oil}}$

G_{P2}	Electricity
Coal	$p_{\text{Electricity,Coal}}$ $+VC_{\text{Truck,Electricity,Coal}} * VP_{Truck}$ $+VC_{\text{Barge,Electricity,Coal}} * VP_{Barge}$

Figure 8: Graphical representation of the group G_P containing the result of multiplication of G_{P1} by G_C and the result of the projection of G_P into G_{P2}.

The multiplication between G_{P1} and G_C gives us a group of processes G_P where the processes of G_{P1} are multiplied by the coefficients of G_C. Then, this group is projected into the destination group G_{P2}, where the already existing processes values are summed to the values of G_P. Figure 8 shows the group G_P and the group G_{P2} after the whole operation.

4.1 Multiplication between the upstream group and the group of coefficients

An obvious method for the first step is a natural join (denoted by $\bowtie$ in relational algebra) between the two groups and a multiplication between the processes impacts and the coefficients values. But F and F' can have different dimensions: not the same number or a not null covering while dimensions are different. (i.e.: if $D_i \cap D'_{i'} \neq \emptyset$ with $i \in \{1, \ldots, n\}$ and $i' \in \{1, \ldots, n'\}$ we could still have $D_i \neq D'_{i'}$). Thus a natural join is not sufficient. We have to specify which dimensions need to be part of the join and do an equijoin. We define the notion of matching set as follows:

Definition 5. Let the matching set $A(F, F')$ be the set containing ordered pairs of dimensions for groups $F = (\mathscr{D}, V)$ and $F' = (\mathscr{D}', V')$ such that:

$$A(F, F') = \{(D, D') \in \mathscr{D} \times \mathscr{D}' \mid D \cap D' \neq \emptyset\} \qquad (2)$$

Example of matching set for two groups: we have two groups F and F' such that $F = (\{D_1, D_2\}, V)$, $F' = (\{D'_1, D'_2\}, V')$, $D_1 = \{A, B\}$, $D_2 = \{C, D\}$, $D'_1 = \{C, D, E\}$ and $D'_2 = \{B, H\}$. Thus, the matching set is $A(F, F') = \{(D_1, D'_2), (D_2, D'_1)\}$.

Having computed the matching set, we can use an equijoin between F and F' and multiply their values (the equijoin operation is denoted by $\bowtie_{[C]}$ where C is the condition). We introduce a new operator named 'Π-join' between two groups for this operation.

Definition 6. Let F and F' be two groups. Let A be the matching set between F and F', we denote by $F \bowtie_{\Pi} F'$ the Π-join between F and F' such that:

$$F \bowtie_{\Pi} F' = \Pi_{[D_{p1},\ldots,D_{pn},V*V']}(F \bowtie_{[A]} F') \tag{3}$$

The condition of the equijoin between F and F' is created using the matching set A such that the first element of each ordered pair in A is equal to the second element. For instance, assuming we have a matching set $A = \{(D_1, D_1'), (D_2, D_3')\}$, the condition is "$D_1 = D_1' \& D_2 = D_2'$".

We denote by P the result of this operation, the dimensions set of P $(D_{p1}, \ldots, D_{pn})$ is denoted by $\mathscr{D}_P$.

Definition 7. Let $F = \{\mathscr{D}, V\}$ and $F' = \{\mathscr{D}', V'\}$ be two groups, let A be the matching set between those two groups, we denote by $\mathscr{D}_A$ the set of F's dimensions which belongs to A and by $\mathscr{D}_A'$ the set of F''s dimensions belonging to A. We denote by $\mathscr{D}_P$ the set of dimensions of the Π-join result between F and F' such that:

$$\mathscr{D}_P = (\mathscr{D} - \mathscr{D}_A) \cup (\mathscr{D}' - \mathscr{D}_A') \cup (\mathscr{D}_A \cap \mathscr{D}_A') \tag{4}$$

We define the intersection between dimensions sets as follows:

$$\mathscr{D}_1 \cap \mathscr{D}_2 = \{D_1 \cap D_2 \mid D_1 \in \mathscr{D}_1 \wedge D_2 \in \mathscr{D}_2 \wedge D_1 \cap D_2 \neq \emptyset\} \tag{5}$$

Using relational algebra, the matching set for two groups F and F' and the sets $\mathscr{D}_A$ and $\mathscr{D}_A'$ are determined as follows:

$$\begin{array}{c} \forall d \in \Pi_{Dim}(\sigma_{(Group=F)} GroupsDimensions), \\ \forall d' \in \Pi_{Dim}(\sigma_{(Group=F')} GroupsDimensions), \\ \text{if } (\Pi_{Keyword}(\sigma_{(Dim=d)} Dimensions) \cap \\ \Pi_{Keyword}(\sigma_{(Dim=d')} Dimensions) \neq \emptyset) \\ \text{then } (d, d') \in A, d \in \mathscr{D}_A \text{ and } d' \in \mathscr{D}_A' \end{array} \tag{6}$$

This is equivalent to the following pseudo-code using SQL queries (in the pseudo-code snippets the relation *Dimensions(Dim, Keyword)* is replaced by the relation *D(Dm, Kw)* and *GroupsDimensions(Group, Dim)* is replaced by *GP(Gp, Dm)*):

```
MatchingSet (F1,F2,pairs,Da1,Da2)
  {Assuming we have a data structure to store
   ordered pairs of dimensions with an add method
   and a data structure to store single dimension
   with an add() method}
 For all d1 in 'SELECT Dm FROM GP WHERE Gp=F1' do
   For all d2 in 'SELECT Dm FROM GP WHERE Gp=F2' do
     If 'SELECT COUNT(D1.Kw)
         FROM Dimensions D1
         INNER JOIN D D2 ON D1.Dm=d1
                          AND D2.Dm=d2
                          AND D1.Kw=D2.Kw' != 0
       pairs.add((d1, d2))
       Da1.add(d1)
       Da2.add(d2)
     Endif
```

The following function determine P's dimensions.

```
CalculateDp (F1,F2,Da1,Da2,Dp)
  {Assuming we have a data structure to store a
   dimension set with an add() method, union
   and minus set-operators and an intersection
   operator as defined in equation (5)}
  For all d in 'SELECT Dm FROM GP WHERE Gp=F1'
    DF1.add(d)
  EndForall
  For all d in 'SELECT Dm FROM GP WHERE Gp=F2'
    DF2.add(d)
  EndForall
  Dp = (Da1 INTERSECT Da2)
       UNION (DF1 - Da1) UNION (DF2 - Da2)
End
```

For instance, consider two groups $F1$ and $F2$ such that $F1 = (D1, D2, V)$, $F2 = (D1, D2, V)$, $F1.D1 \cap F2.D1 \neq \emptyset$ and $F1.D2 \cap F2.D2 \neq \emptyset$. The SQL corresponding to $F1 \bowtie_{\Pi} F2$ is as follows:

```
SELECT F1.D1, F1.D2, F1.V*F2.V
  FROM F1, F2
  WHERE F1.D1=F2.D1 AND F1.D2=F2.D2;
```

4.2 Projection into the destination group

The second step is the projection of the result P into the destination group F'', this operation is almost identical to the Π-join, the matching set is determined between P and F''. We introduce a new operator named 'Σ-join' between two groups for this operation.

Definition 8. Let F and F' be two groups, let A be the matching set between F and F', we denote by $F \bowtie_{\Sigma} F'$ the Σ-join between F and F' such that:

$$F \bowtie_{\Sigma} F' = \Pi_{[D_1,\ldots,D_n,V+V']}(F \bowtie_{[A]} F') \tag{7}$$

Consider two groups P and F such that $P = (D1, D2, V)$, $F = (D1, D2, V)$, $P.D1 \cap F.D1 \neq \emptyset$ and $P.D2 \cap F.D2 \neq \emptyset$. The SQL corresponding to $P \bowtie_{\Sigma} F$ is as follows:

```
SELECT F1.D1, F1.D2, F1.V+F2.V
  FROM F1, F2
  WHERE F1.D1=F2.D1 AND F1.D2=F2.D2;
```

This is feasible only if P and F'' have the same number of dimensions. If P has at least one dimension not in the matching set we have to aggregate P's values which are indexed by the dimension not in F''. This case occurs if $\overline{\mathscr{D}_A^P} \neq \emptyset$ where $\overline{\mathscr{D}_A^P} = \mathscr{D}_P - \mathscr{D}_A^P$ and $\mathscr{D}_A^P$ are the common dimensions between P and F'' (determined while calculating the matching set between P and F''). Thus, we have to create the group P_{agg} such that ($G_{Sum()}$ is the aggregation operation sum):

$$P_{\text{agg}} = {}_{\mathscr{D}_A^P}G_{\mathscr{D}_A^P,Sum(V)}(P) \tag{8}$$

For instance, consider two groups P and F such that $P = (D1, D2, D3, V)$, $F'' = (D1, D2, V)$, $F1.D1 \cap F2.D1 \neq \emptyset$ and $F1.D2 \cap F2.D2 \neq \emptyset$. In order to calculate $P \bowtie_{\Sigma} F$ we have to calculate P_{agg}. The SQL query to calculate P_{agg} is as follows:

```
SELECT D1, D2, SUM(V)
  FROM P
  GROUP BY D1, D2;
```

4.3 Different types of macro-relations

To define a new macro-relation we have to specify the upstream group, the destination group and the group of coefficients. We can distinguish four types of composition relations between groups of processes depending on the three groups specified in the macro-relation:

- **Normal relation:** the three groups have the same dimensions[3]. This is the most common type of macro-relation.
- **Partial relation:** the three groups have the same number of dimensions but at least one (not null) intersection of dimensions between groups is a smaller set than one of the dimensions of the intersection. This type of relation is useful if we want to use a group in several relations and we do not want to model a relation with every process in the group.
- **Aggregation relation:** the destination group has fewer dimensions than the result of the multiplication of the upstream group and the group of coefficients. If we want to model relations between a process and all its upstream processes, we can create a group containing all the upstream processes and create a relation between the studied process and this group.
- **Projection relation:** the destination group has more dimensions than the result of the multiplication of the upstream group and the group of coefficients. This is useful to factorize the relation between one process and multiple processes: the destination processes are grouped and we create a relation between the upstream process and this group (this also works if there are multiple upstream processes).

Furthermore, we can combine the different types of macro-relations to obtain more expressive relations like: aggregation and projection, partial aggregation, etc. The next section illustrates some of these macro-relations types.

5. CASE STUDY: ELECTRICITY PRODUCTION IN THE U.S.

The National Renewable Energy Laboratory (NREL) provides a LCI Database containing inventory data for electricity production for every 27 U.S. subregions[2].

These subregions are defined by the U.S. EPA's Emissions and Generation Resource Integrated Database (eGRID)[5]. An eGRID subregion represents a portion of the US power grid that is contained within a single North America Electric Reliability Council (NERC) region, and generally represents sections of the power grid which have similar emissions and resource mix characteristics, and may be partially isolated by transmission constraints.

The NREL's LCI data could be exported in the ecospold format[16] as Excel spreadsheets. This is a common data exchange format widely used in the LCA community. Besides including some meta-data for each process (provenance, comments, etc.), it includes dependencies between processes with the dependency coefficients. Some processes in this database are not detailed (and flagged as ”dummy” processes), thus we do not have a complete data set in terms of dependency relations. But, even after pruning these dummy processes, the data set is still complex enough to illustrate our proposition. Figures 9 and 10 show, respectively, the detailed-graph and the macro-graph corresponding to this data set limited to 7 subregions.

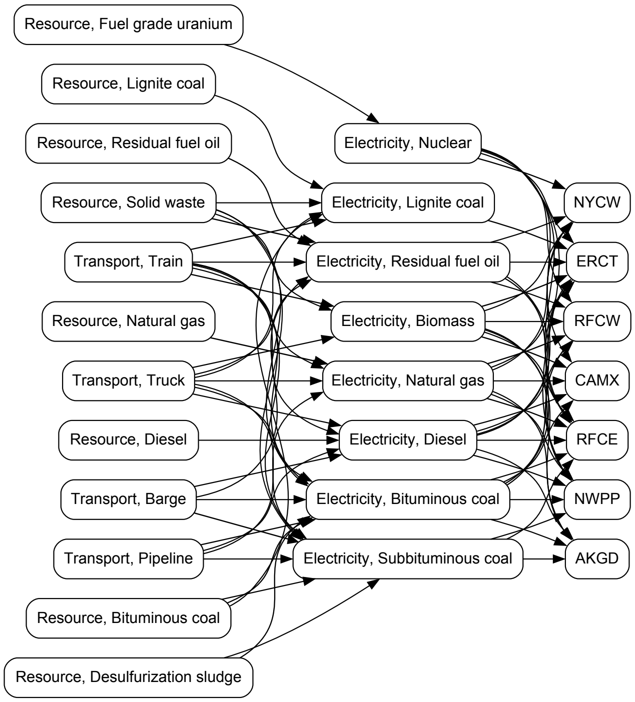

Figure 9: Sequence from the studied data set extracted from the NREL's LCI Database.

Restricting the data set to 7 eGRID subregions, we have 27 processes and 72 inter-processes relations in the detailed-graph. Whereas, in the macro-graph, we have 13 groups of processes and 17 macro-relations, thus we have 17 groups of coefficients. But, using other types of macro-relations than the normal relation, we can reduce the number of groups of coefficients to 12. For instance, the *Transports* group of processes can be linked with partial relations to every electricity group of processes using a group of coefficients containing all the dependency coefficients between transportation systems processes and the electricity production processes (in a similar way we illustrate the translation procedure as shown in Figure 8). We could even use only 8 groups of coefficients if we store all the dependency coefficients between every electricity production group of processes to the eGRID group of processes.

Illustrations for some types of macro-relations

The group containing transportation system processes encompasses processes for four transportation systems: train, barge, truck and pipeline. The electricity production from the biomass group of processes has only one process (it is in fact a single process). Thus the transport processes are multiplied by the dependencies coefficients between trans-

[3] Process uniqueness is guaranteed by its identifier (i.e.: its keywords), under such circumstances, processes indexed in the upstream and the destination groups are the same. We use common keywords to differentiate groups having the same dimensions. Those keywords behave like a dimension with a single keyword.

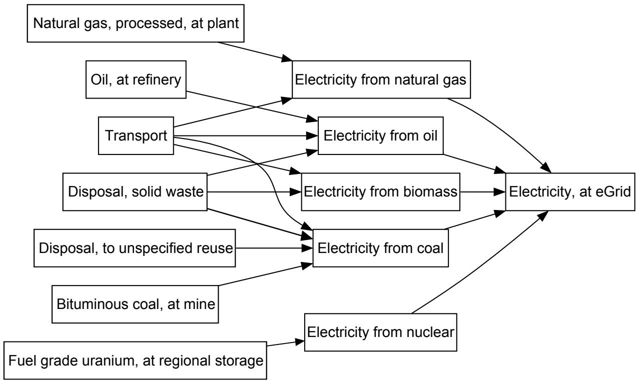

Figure 10: The macro-graph of the studied data set extracted from the NREL's LCI Database.

port and biomass and are summed into the biomass process. In the macro-graph we have only one directed edge from the transport processes group to the electricity from biomass process group. This macro-relation is translated, in the detailed-graph, into four directed edges from the four transportation processes to the electricity production from biomass process. Thus we have created an aggregation macro-relation.

Not every eGRID subregion uses nuclear power plant. In order to create a macro-relation from the nuclear electricity production group of processes to the eGRID electricity production group of processes, we need to create a partial relation (in fact it's a partial projection). Even if the latter group contains only one process, we consider it as a group to have a systematic way of dealing with the translation. The group of coefficients between those two process groups has a dimension containing only the eGRID subregions using nuclear power plants. On our subset, the Alaska subregion does not rely on nuclear energy, so the macro-relation is translated into micro-relations between the nuclear electricity production process and every eGRID subregion electricity production process except for Alaska.

6. CONCLUSIONS AND FUTURE WORK

We proposed a new approach to model LCI using an ontology and relations between semantic groups of processes. The key benefit of this approach is to offer a more understandable model of LCI databases. This approach also provides an ontology driven way to create new relations between processes.

For performance reasons, we implemented our model using relational algebra and SQL. In the future we plan to use OWL[12] with description logic rules and a semantic reasoner, and eventually study the impact performance of the system on big energy data sets.

Processes indexation using keywords stored in an ontology can also be used to answer queries like: what is the impact of processes indexed with a keyword (or a set of keywords) on a specific process. For instance we could get the impacts of transport processes on the electricity production for a specific eGRID subregion. This can be done by restricting the calculation to upstream processes indexed with a specific keyword. We can achieve this goal using SPARQL, or a modified version of SPARQL because of some limitations on the property path functionality.

7. REFERENCES

[1] GABI Life Cycle Inventory Databases, http://www.gabi-software.com/ last accessed: 16/12/2011. PE International.

[2] U.S. Life Cycle Inventory Database, http://www.nrel.gov/lci/ last accessed: 01/09/2011. National Renewable Energy Laboratory.

[3] ISO 14044 (2006): Environmental Management – Life Cycle Assessment – requirements and guidelines. International standard, International Organisation for Standardisation (ISO), 2006.

[4] D. Beckett. Turtle Terse RDF Triple Language. W3C Recommendation, World Wide Web Consortium, 2011.

[5] E. H. Pechan & Associates., Inc. The Emissions & Generation Resource Integrated Database for 2010 Technical Support Document. U.S. Environmental Protection Agency, Washington, D.C., 2010.

[6] R. Frischknecht, G Rebitzer. The ecoinvent database system: a comprehensive web-based LCA database. *Journal of Cleaner Production*, 2005.

[7] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 1993.

[8] J. B. Guinée. *Handbook on Life Cycle Assessment: Operational Guide to the ISO Standards.* Springer, New York, 2002.

[9] R. Heijungs, S. Suh. *The computational structure of life cycle assessment.* Kluwer Academic Publishers, Dordrecht, 2002.

[10] G. Klyne and J. J. Carroll editors. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, World Wide Web Consortium, 2004.

[11] W. Leontief. *Input-Output Analysis*, pages 53–83. University of British Columbia Press, 1986.

[12] D. McGuinness and F. van Harmelen editors. OWL 2 Web Ontology Language Document Overview. W3C Recommendation, World Wide Web Consortium, 2009.

[13] D. L. McGuinness. *Ontologies come of age*, page 171–192, in: The Semantic Web: Why, What, and How. MIT Press, 2002.

[14] G. P. Peters. Efficient Algorithms for Life Cycle Assessment, Input-Output Analysis, and Monte-Carlo Analysis. *The International Journal of Life Cycle Assessment*, 2007.

[15] E. Prud'hommeaux A. Seaborne editors. SPARQL Query Language for RDF. W3C Recommendation, World Wide Web Consortium, 2008.

[16] R. Frischknecht, N. Jungbluth., H. Althaus, et al. Introduction The ecoinvent Database: Overview and Methodological Framework. *The International Journal of Life Cycle Assessment*, 2005.

[17] R. Varga. *Matrix Iterative Analysis.* Springer Series in Computational Mathematics. Springer Verlag, 2010.

[18] WK Nicholson. *Elementary linear algebra with applications.* PWS-Kent Publishing Company, 1990.