Life cycle fragments: development of an online tool for curating and sharing life

Brandon Kuczenski
University of California, Santa Barbara, bkuczenski@ucsb.edu

International Environmental Modelling and Software Society (iEMSs)
7th Intl. Congress on Env. Modelling and Software, San Diego, California, USA
Daniel P. Ames, Nigel W. T. Quinn and Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

Life cycle fragments: computational methods for analyzing and sharing life cycle assessment models

Brandon Kuczenski [a]

[a] *Institute for Social, Behavioral, and Economic Research (ISBER)*
University of California, Santa Barbara, CA USA 93106-2150 (bkuczenski@ucsb.edu)

Abstract: Increasingly, policy makers and consumers demand information on the environmental implications of industrial activities. Life cycle assessment (LCA) is a standardized methodology to relate the delivery of products and services to the potential environmental impacts from that delivery, both directly and throughout the industrial supply chain. LCA requires extensive information about industrial processes throughout the global economy, and is subject to substantial parametric and epistemic uncertainty. Moreover, results often hinge on modeling decisions, such as the selection of system boundaries and data sources, leading to complications when validating results or comparing study outcomes. Comparative results from a single study can give policy-relevant insights only if it is possible to review the sensitivity of results to both uncertainty and modeling decisions. After completing an LCA study for the state of California to inform policy on waste lubricating oil management, we were motivated to develop a data framework for formally describing life cycle inventory models and allowing them to be inspected and analyzed. A “life cycle fragment” describes a network of dependencies among industrial processes with a limited scope and simple formal structure, simplifying the computation of LCA results. Efforts to construct an online tool implementing the fragment framework for use by policy makers, stakeholders, and the general public are ongoing. The tool will facilitate use of the model to analyze parametric uncertainty and model design, thus improving transparency and communicability of the model’s results to interested parties.

Keywords: Life cycle assessment; sensitivity analysis; confidentiality of data; environmental policy development.

1 INTRODUCTION

1.1 Life Cycle Assessment

Estimating the environmental impacts of an activity requires accounting for the environmental burdens that result directly from that activity, as well as the indirect burdens associated with producing, distributing, and disposing of products required for the activity to be conducted. Life cycle assessment (LCA) is a standardized methodology for estimating the total environmental implications of products or services. LCA is governed by a family of international standards, the most prominent of which is ISO 14044 [2006]. The same principles that underlie ISO 14044 also guide numerous other related methods such as environmentally-extended input-output LCA [Hendrickson et al., 1998; Wiedmann, 2009]; ecological footprint [Wackernagel et al., 1999]; and the PAS 2050 carbon footprint standard [PAS 2050, 2011].

An array of fundamental uncertainties pervade current techniques for conducting LCA [Williams et al., 2009]. LCA results are sensitive to parameter values in descriptions of industrial processes, to the inclusion or omission of relevant environmental interventions (such as emissions), and characterization factors describing the degree of impact on the environment from a given intervention. However, LCA results also depend to a great degree on design decisions by LCA modelers, including the selection of a system's scope and boundary, the choice of data providers or data sets, and other model-specific choices. As a consequence, it is difficult to draw scientific generalizations from LCA results. More general findings require a heavily interpretive process of meta-analysis, in which study assumptions and modeling decisions are reconciled (e.g. Lifset [2012]). The result is an array of "unstable conclusions" [Henriksson et al., 2014] that undermine public confidence in LCA results.

Before LCA-based findings can be brought to bear on environmental policy, it is necessary to allow public stakeholders to fully inspect and evaluate both the study methods and results. We were motivated by a recent policy-relevant LCA investigation [Geyer et al., 2013] to develop a framework for publicizing LCA results in an interactive format that would (1) permit members of the public to explore the study model and evaluate its sensitivity to input parameters, and (2) allow the public agency responsible for the study to update and revise the model as new data became available, without requiring significant outside expertise. The framework will be instantiated as an on-line tool implementing the original study and hosted by the agency that funded the study.

1.2 LCA Computation

ISO-style LCA describes the delivery of a product or service as a network of industrial processes whose outputs are required in order to provide a "functional unit" of utility to a user. For example, to provide the service of "laundering 1 dry cubic meter of clothing" may require operating a washing machine and clothes dryer for several cycles, leading to direct consumption of water, electricity, and detergent. It therefore also requires the extraction of coal and its transport to a power plant to produce the electricity; the manufacture of chemical precursors to the detergent, and so on. Depending on how the system is constructed, it may also require the allocation of some of the impacts from manufacturing the washing machine and dryer, and of disposing them at the end of their useful life. This last step may actually lead to apparently negative impacts because it results in the generation of steel scrap, which may be used to avoid or "displace" the production of primary steel.

Computing an LCA has two main quantitative steps: life cycle inventory preparation, in which the set of environmental interventions is identified; and life cycle impact assessment (LCIA), in which those interventions are characterized and summed to quantify potential environmental impacts. Subsequent interpretive efforts, such as contribution analysis, sensitivity analysis, and scenario modeling, repeat those steps using perturbed models and/or data.

The general LCA computation can be summarized as a sequence of matrix operations. The life cycle inventory model includes a set of n processes related by an $n \times n$ matrix of "direct requirements" A. Element a_{ij} represents the activity of process i required to produce a unit outpt of process j. The "total requirements" matrix $(I - A)^{-1}$ reports the combination of direct and upstream requirements of each process in columns. The n-element output vector $\mathbf{y}$ describes the net output implied by the functional unit; e.g. the quantity of laundry that must be done. Thus $(I - A)^{-1} \cdot \mathbf{y}$ yields a vector of process activity levels required to achieve the functional unit.

Then, for m distinct recognized environmental interventions, $m \times n$ matrix B lists the interventions that result from a unit activity of each process. Finally, for t recognized impact categories, $t \times m$ matrix E lists the characterization of each environmental intervention with respect to each category. Then a t-element vector of impact scores $\mathbf{s}$ can be computed as [Suh and Huppes, 2005]:

$$\mathbf{s} = E \cdot B \cdot (I - A)^{-1} \cdot \mathbf{y} \tag{1}$$

2 APPROACH

2.1 Life Cycle Fragments

To carry out the project, it was necessary to develop a novel computing framework to represent the structure of life cycle inventory models. We describe such a model as a directed graph whose arcs indicate dependency relationships between processes, denoted by nodes. An arc indicates a reliance by the head node, called the parent, upon an output produced by the tail node, called the dependency. Each arc includes a weight that indicates the quantity of input associated with a unit of activity of the parent node. This quantity is sometimes known as an “exchange”. The set of a node’s dependencies together with their weights is called its *inventory* and corresponds to a column in the A matrix above.

We introduce a *life cycle fragment*, which is a subgraph of such a graph that can be fully covered by a spanning tree. In other words, no node can have no more than one parent node.[1] A fragment $F = \{V, D, w, B\}$, in which V is an ordered set of nodes, D is a set of arcs, $w : D \to \mathcal{R}$ is a mapping of arcs to arc weights (quantities), and $B : V \to \mathcal{R}^m$ is a mapping of nodes to environmental interventions, equivalent to the B matrix of Equation 1.

Each node in a fragment can have many outputs and dependencies, and not all of them must be resolved within the fragment. A fragment determines a set of input and output *flows*: respectively, dependency relationships not satisfied and coproducts not utilized in the fragment. Note that coproducts that *are* used within the fragment effectively denote dependencies on the consuming systems being available to receive them.

The characteristic property of a fragment is its *reducibility* to a single node whose characteristics with respect to the LCA computation are identical to the original graph. This *reduction* can be performed through successive translation of dependency relationships to parent nodes. This is depicted in Figure 1. A fully reduced fragment is a star graph. As a consequence of fragment reducibility, a fragment’s direct requirements matrix can be represented in the following *canonical form*:

$$\hat{A} = \begin{pmatrix} 0 & \cdots & 0 & a_1 \\ 0 & \cdots & 0 & a_2 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & a_k \\ 0 & \cdots & 0 & 0 \end{pmatrix}; \quad \hat{x} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \\ 1 \end{pmatrix} \tag{2}$$

Here each a_k is a product of one or more arc weights, depending on the fragment’s structure. $\hat{x}$ is the rightmost column of the matrix $(I - \hat{A})^{-1}$ and thus indicates the activity levels of each node corresponding to a unit activity level of the primary output (i.e. the tree’s root node). The impact score can thus be written $\mathbf{s} = E \cdot B \cdot \hat{x}$.

2.2 Parameterization

A dependency relationship in the unreduced graph can be varied parametrically with the representation $a_{ij}(1 + \phi)$. Assuming all other relationships are unchanged, the $(1 + \phi)$ factor will be preserved through the successive multiplications carried out during graph reduction. As a consequence, every dependency relationship affected by the parameterization will include exactly one $(1 + \phi)$ term. Thus

[1] It is also possible to extend the development discussed here to include any directed acyclic graph.

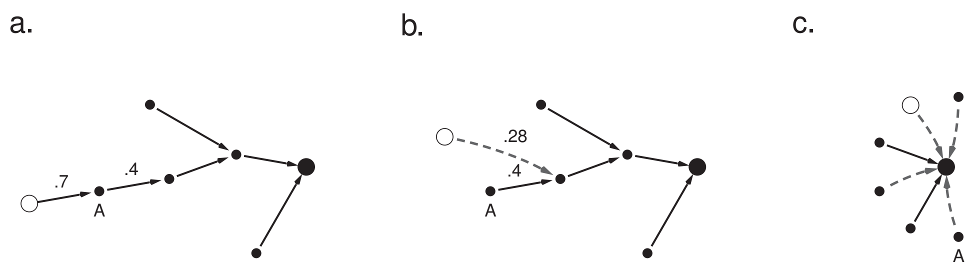

Figure 1. Reduction of a life cycle fragment to canonical form by successive dependency translation. (a) A fragment represented as a dependency tree network. Weights 0.7 and 0.4 are shown for two successive dependencies around the node labeled 'A.' (b) Node A's dependency on the open-circle node has been translated to its parent node, where it is indicated by a dashed line. The weight of the new dependency is the product of the two prior weights. (c) The fully reduced dependency tree.

the activity vector $\tilde{x}$ under parameterization ϕ can be written:

$$\tilde{x} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \\ 1 \end{pmatrix} + \begin{pmatrix} \cdot \\ a_f \\ \cdot \\ a_g \\ \cdot \\ 0 \end{pmatrix} \cdot \phi = \hat{x} + \tilde{x}_\phi \phi \tag{3}$$

where the second term includes only coefficients from the set $\{a_1 \dots a_k\}$ where the parameterization term appears, and zero elsewhere. Note that parameterizations in distinct branches of the tree that do not interact can be additively combined:

$$\tilde{x} = \hat{x} + \tilde{x}_{\phi_1}\phi_1 + \tilde{x}_{\phi_2}\phi_2 + \dots \tag{4}$$

The overall impact score is computed linearly from the parameterized activity levels:

$$\tilde{\mathbf{s}} = \mathbf{s} + \mathbf{s}_{\phi_1}\phi_1 + \mathbf{s}_{\phi_2}\phi_2 + \dots \tag{5}$$

in which $\mathbf{s}_{\phi_k} = E \cdot B \cdot \tilde{x}_{\phi_k}$ is straightforwardly the *sensitivity* of the result to parameter ϕ_k. Note that a similar construction can be trivially applied to the B and E matrices.

3 Technical Implementation

3.1 Online Tool Requirements

For a given LCA study, the underlying inventory model is carefully crafted by an LCA practitioner to meet the goal and scope of a study. However, the modeling decisions and strategies employed may not always be easily intelligible to non-experts. The current project was conceived as a way to communicate model structure and sensitivity as well as results to the reporting audience, as a complement to a technical report.

Based on the interests of the funding agency and the stakeholders, the following requirements were identified:

1. *Model Inspection.* The user must be able to view the model structure and inspect the value of quantitative parameters that determine its output.
    - Enable representation of the inventory model as a directed graph.

- Each node on the graph represents a unit process in the inventory model, reporting dependencies on other processes and environmental interventions.
- Impact Characterization Factors should be shown or downloadable.

2. *Presentation and Analysis of Results.* The life cycle impact indicator scores must be viewable, and their derivations shown. The interface should support:
   - *Contribution analysis*, showing the relative contributions of different parts of the model to the total results.
   - *Sensitivity analysis*, showing the effect of the changes in user-selectable input parameters on the total results.
   - *Scenario analysis*, showing side-by-side comparisons of comparable models.
3. *Protection of Confidential Data.* Inventory data provided by stakeholders or used under license from data providers should not be visible or extractable using the tool.
4. *Export.* Quantitative results should be exportable in standard data formats, particularly the emerging ILCD framework.
5. *Data Updates.*
   - Privileged users should be able to upload alternate source data sets for some or all components.
   - The tool should aid in reporting the differences between original and alternate data sets and the influence of the alternate data on model results.

3.2 The ILCD Format

As LCA increases in prominence, particularly in Europe there has been significant effort to normalize its use and ensure a consistent methodology. The International Reference Life Cycle Data (ILCD) Format has been developed by the Joint Research Centre of the European Commission to advance this objective [Recchioni et al., 2013]. The format provides a formal specification for interchange of ISO 14044-style LCA data. It is paired with an online database (known as ELCD) that contains validated reference data. There has been a recent convergence around the ILCD format, and all major LCA software tools, as well as several prominent data repositories, now support it. The format includes data types for *processes* and *flows* (together defining the A and B matrices above), and *LCIA methods* (defining the E matrix), plus four other data types that predominantly report metadata [Wolf et al., 2011].

3.3 Application Design

Figure 2 shows the architecture of the fragment tool. The ILCD format is used as the core of the database schema in order to foster data exchange with other tools in the future. The fragment implementation extends the ILCD format with network modeling to support the computations discussed above. The application front-end interacts with the database via a REST API, enabling staged implementation and straightforward extensibility [Fielding and Taylor, 2002].

The tool has three distinct user bases. First, the general public is provided with the ability to inspect dependency models already designed, view results, and investigate parameter sensitivity. Because fully reduced fragments are equivalent to processes, dependency models can be nested and viewed hierarchically. Users can select a set of parameters of interest and the tool will propagate their effects through the model. Second, the tool administrative interface provides the capacity update and compare data sets, backup, and perform similar management tasks. Third, the design interface permits model designers to directly construct fragments by successively resolving dependencies. Designers can also construct and store model configurations (parameter sets) that can be accessed by the public.

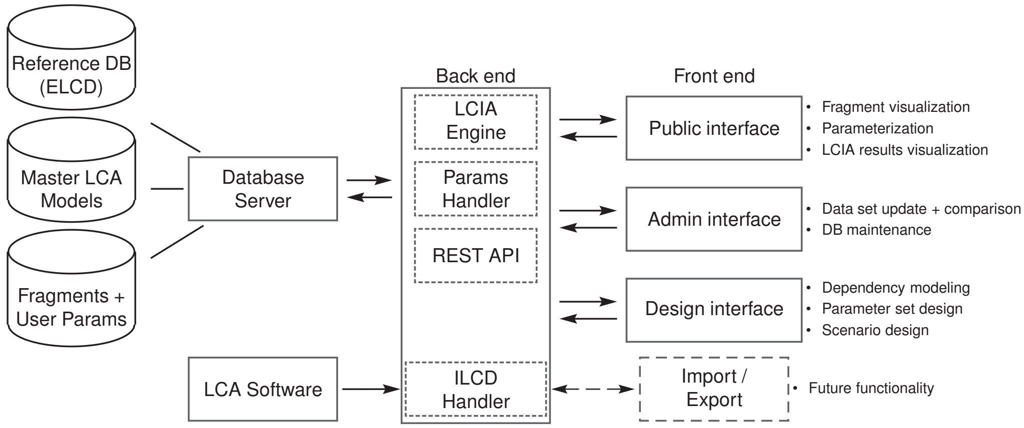

Figure 2. Architecture of the fragment tool.

3.4 LCIA Computation

Because both the B and E matrices are sparse, the computation in Equation 1 is implemented using database operations rather than by matrix multiplication. Each vector product implied in $E \cdot B$ is represented as an inner join, on elementary flows, of an LCIA method table and a process inventory table. For a finite set of user-selected parameters, the sensitivities can be represented as in Equation 5 and all sensitivity terms can be calculated and transmitted to the frontend all at once. The user can subsequently interact with the model, adjust parameters and perform the analyses detailed in Section 3.1 without having to communicate further with the backend or database server.

4 Outlook and Significance

Though designed for a single application, the computational model embodied in the fragment tool promises to improve the documentation and communication of life cycle inventory models generally. A fragment can be described formally as a network of "process" and "flow" entities as defined in the ILCD standard, and thus can be implemented as an extension of that standard. Reduction effectively converts a fragment into a process. Thus a fragment can be both constructed from and cast into ILCD entity types.

It is generally accepted that inventory information (arc weights in A and elements or columns of B) considered to be confidential are adequately concealed by the computation of $B \cdot x$ (though at the expense of limiting analysis). However, in some cases it may be possible to expose an unweighted dependency graph without compromising confidentiality. Thus fragments could improve the transparency of inventory models without revealing confidential data and can support formal reasoning about confidentiality.

The fragment representation satisfies the analytic requirements listed in item 2 in Section 3.1 and thus contributes to ongoing efforts to improve the state of LCA meta-analysis. First, representation of $\hat{x}$ from Equation 2 as a diagonal matrix will expose a contribution analysis over the nodes of the fragment. Sensitivity analysis is demonstrated by fragment parameterization. And scenario analysis can be implemented by generating alternative dependency graphs or parameter sets with otherwise equivalent contribution and sensitivity analysis.

ACKNOWLEDGMENTS

The author thanks Robert Carlson (CalRecycle) for managing the project and providing design input; Kyle Meisterling (UCSB) for conceptual development; and the anonymous reviewers. The work is supported by CalRecycle Contract #DRR13026.

REFERENCES

Fielding, R. T. and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Trans. Internet Technol.*, 2(2):115–150.

Geyer, R., Henderson, A., Kuczenski, B., and Zink, T. (2013). Life Cycle Assessment of Used Oil Management in California pursuant to Senate Bill 546 (Lowenthal). Technical report, CalRecycle.

Hendrickson, C., Horvath, A., Joshi, S., and Lave, L. (1998). Economic input-output models for environmental life-cycle assessment. *Environ. Sci. Technol. A*, 32(7):184A–191A.

Henriksson, P. J. G., Guine, J. B., Heijungs, R., Koning, A., and Green, D. M. (2014). A protocol for horizontal averaging of unit process dataincluding estimates for uncertainty. *The International Journal of Life Cycle Assessment*, 19(2):429–436.

ISO 14044 (2006). *Environmental management — Life cycle assessment — Requirements and guidelines*. ISO, Geneva, Switzerland.

Lifset, R. (2012). Toward meta-analysis in life cycle assessment. *Journal of Industrial Ecology*, 16:S1–S2.

PAS 2050 (2011). Specification for the assessment of the life cycle greenhouse gas emissions of goods and services. Technical report, BSI / DEFRA.

Recchioni, M., Mathieux, F., Goralczyk, M., and Schau, E. M. (2013). Ilcd data network and elcd database: current use and further needs for supporting environmental footprint and life cycle indicator projects. Technical Report EUR 25744 EN, European Commission, Joint Research Centre, Luxembourg.

Suh, S. and Huppes, G. (2005). Methods for life cycle inventory of a product. *Journal of Cleaner Production*, 13(7):687–697.

Wackernagel, M., Onisto, L., Bello, P., Linares, A. C., Falfn, I. S. L., Garcia, J. M., Guerrero, A. I. S., and Guerrero, M. G. S. (1999). National natural capital accounting with the ecological footprint concept. *Ecological Economics*, 29(3):375 – 390.

Wiedmann, T. (2009). A review of recent multi-region inputoutput models used for consumption-based emission and resource accounting. *Ecological Economics*, 69(2):211 – 222. Special Section: Analyzing the global human appropriation of net primary production - processes, trajectories, implications.

Williams, E. D., Weber, C. L., and Hawkins, T. R. (2009). Hybrid framework for managing uncertainty in life cycle inventories. *Journal of Industrial Ecology*, 13(6):928–944.

Wolf, M.-A., Düpmeier, C., and Kusche, O. (2011). The international reference life cycle data system (ILCD) format–basic concepts and implementation of life cycle impact assessment (LCIA) method data sets. In *Proceedings of the 25th EnviroInfo Conference Innovations in Sharing Environmental Observations and Information*. Schaker-Verlag.