



Netzwerk Lebenszyklusdaten

Arbeitskreis METHODIK

AP 3 PARAMETRISIERUNG

Projektbericht

im Rahmen des Forschungsvorhabens FKZ 01 RN 0401 im Auftrag
des Bundesministeriums für Bildung und Forschung

Ifu Hamburg GmbH
Hochschule Pforzheim
PE INTERNATIONAL GmbH

Hamburg Pforzheim Leinfelden-Echterdingen Karlsruhe - Juni 2007

ifu Hamburg GmbH

HOCHSCHULE PFORZHEIM



PE INTERNATIONAL
EXPERTS IN SUSTAINABILITY

Hrsg.: Forschungszentrum Karlsruhe
Institut für Technikfolgenabschätzung und Systemanalyse –
Zentralabteilung Technikbedingte Stoffströme



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft

Vorwort

Der vorliegende Projektbericht wird herausgegeben vom Netzwerk Lebenszyklusdaten (www.netzwerk-lebenszyklusdaten.de).

Das Netzwerk Lebenszyklusdaten ist die gemeinsame Informations- und Koordinationsplattform aller in die Bereitstellung und Nutzung von Lebenszyklusdaten in Deutschland involvierten Gruppen – von Wissenschaft und Wirtschaft über Politik und Behörden hin zu Verbraucherberatung und allgemeiner interessierter Öffentlichkeit. Ziel des Netzwerks Lebenszyklusdaten ist es, das umfangreiche Knowhow auf dem Gebiet der Lebenszyklusdaten innerhalb Deutschlands zusammenzuführen und als Basis zukünftiger wissenschaftlicher Weiterentwicklung und praktischer Arbeiten für Nutzer in allen Anwendungsgebieten von Lebenszyklusanalysen bereitzustellen.

Das Netzwerk Lebenszyklusdaten wird getragen vom Forschungszentrum Karlsruhe. Die vorliegende Studie wurde im Rahmen der Projektförderung (2004 – 2008) des Bundesministeriums für Bildung und Forschung (BMBF) „Förderung der Wissenskooperation zum Aufbau und Umsetzung des deutschen Netzwerks Lebenszyklusdaten“ erstellt. Weitere im Rahmen dieser Projektförderung erstellte Studien sind erhältlich unter <http://www.netzwerk-lebenszyklusdaten.de/cms/content/Projektberichte>.

Kontakt Netzwerk Lebenszyklusdaten:

E-Mail: info@netzwerk-lebenszyklusdaten.de

Anschrift: Forschungszentrum Karlsruhe GmbH
Institut für Technikfolgenabschätzung und Systemanalyse,
Zentralabteilung Technikbedingte Stoffströme (ITAS-ZTS)
Postfach 3640
76021 Karlsruhe
www.netzwerk-lebenszyklusdaten.de



Das Netzwerk Lebenszyklusdaten wird gefördert durch das
Bundesministerium für Bildung und Forschung



AP 3 – PARAMETRISIERUNG

Autoren:

Prof. Mario Schmidt
Hochschule Pforzheim
Tiefenbronner Str. 65
75175 Pforzheim

Jan Hedemann
Ifu Hamburg GmbH
Große Bergstraße 219
22767 Hamburg

Johannes Kreissig
PE International
Hauptstraße 111-113
70771 Leinfelden-Echterdingen

Inhalt

Inhalt	3
Einführung	4
Zwecke und Einbindung in das Life-Cycle Assessment.....	4
Zwecke der Parametrisierung	5
Grundlagen des Life-Cycle Assessments.....	6
Ökologische Lebenszyklen	9
Zurechnungen	9
Bestände	10
Linearität	11
Berechnungen zur Bestimmung von Life-Cycle Inventories	12
Schlussfolgerungen.....	12
Formen der Parametrisierung.....	14
Parametrisierung mit periodenbezogenen Stoffstrommodellen.....	14
Parametrisierung ohne periodenbezogene Stoffstromanalyse	18
Datenformate für parametrisierte Lebenszyklusdaten	21
Erweiterung vorhandener Datenformate.....	22
Grammatik für Funktionsausdrücke	25
Erweiterungen.....	28
Selbstdefinierte Funktionen	28
Kreuztabellen	31
Resümee.....	32
Literatur.....	34

Einführung

Modelle des Life-Cycle Assessments (LCA) sind spezielle Wirtschaftlichkeitsanalysen. Es geht darum, die negativen Umweltwirkungen von Alternativen im Rahmen der rationalen Entscheidungsfindung abzuschätzen. Sie ergänzen damit die ökonomischen Wirtschaftlichkeitsanalysen, welche insbesondere auf Ansätzen der Kostenrechnung basieren.

Von daher leitet sich die Fragestellung der Parametrisierung nicht unmittelbar aus dem idealisierten Nutzungskontext ab. Die praktische Umsetzung von Konzepten des Life-Cycle Assessments zeigt jedoch, dass sich mit der Parametrisierung eine Reihe von Vorteilen verbindet: Untersuchung verschiedener Szenarien innerhalb einer gegebenen Modellstruktur (Vergleich verschiedener Parameterkonstellationen), Sensitivitätsanalysen und die bessere Anpassbarkeit von standardisierten und in Datenbanken gespeicherten Prozessspezifikationen.

Im Folgenden soll vorgestellt werden, wie sich das Life-Cycle Assessment um die Parametrisierung erweitern lässt, welche Vorteile sich damit verbinden und wie sich dies softwaretechnisch realisieren lässt. Schließlich werden Hinweise gegeben, auf welche Weise Parameter in Lebenszyklusdaten berücksichtigt werden können.

Zwecke und Einbindung in das Life-Cycle Assessment

Veränderbare Eigenschaften eines Modells werden als Modellparameter oder kurz Parameter bezeichnet.

Der Begriff Parameter wird je nach Wissenschaft unterschiedlich verwendet. Allgemein wird er als charakterisierende Eigenschaft beschrieben. In der Informatik unterscheidet man Parameter etwa nach ihrem Einsatzzeitpunkt in einer Software, bspw. Installationsparameter, Konfigurationsparameter und Laufzeitparameter. In der Softwareentwicklung ist ein Parameter ein veränderliche und zugleich charakterisierende Element eines Unterprogramms, das bei der Definition des Unterprogramms formal angelegt (formaler Parameter) und beim Aufruf des Unterprogramms für jeweils diesen einen Aufruf auf einen konkreten Wert gesetzt wird (tatsächlicher Parameter).

Der tatsächliche Parameter definiert die konkrete Bedeutung des abstrakten formalen Parameters. Über den formalen Parameter wird oftmals ein Wertebereich für den tatsächlichen Parameter vorgegeben.

Es ist ein komplexitätsreduzierendes Anliegen in der Informatik, den Algorithmus des Unterprogramms einzig und allein über die explizit angegebenen Parameter zu charakterisieren, d.h. das Ergebnis des Unterprogrammaufrufs ist allein von den Parametern abhängig nicht von anderen Rahmenbedingungen (Vermeidung von Seiteneffekten). Dies hat auch zum Programmiersprachenparadigma der funktionalen Programmierung geführt, die Parameter zulässt, nicht aber veränderliche Variablen, welche die Seiteneffekte möglich machen.

Die Unterscheidung zwischen formalen und tatsächlichen Parametern ist für das Themenfeld LCA Datensätze nicht unerheblich. In einem generischen LCA-Datensatz in einer LCA Datenbank ist es sinnvoll, formale Parameter zu definieren, um den Einsatzbereich des Datensatzes zu spezifizieren und die Grenzen aufzuzeigen. Zugleich wird der Datensatz flexibilisiert und an den Nutzungskontext besser anpassbar. Ebenfalls sinnvoll ist es, einen zulässigen Wertebereich für jeden Parameter vorzugeben. Der Anwender des Datensatzes muss dann einen tatsächlichen Parameter bei Verwendung des Datensatzes festlegen. Im LCA-Bereich ist die Festlegung eines tatsächlichen Parameters nicht immer trivial. Hierzu sollte auch Hilfestellung bspw. durch Vorgabe von Default-Werten gegeben werden.

Im Falle des Life-Cycle Assessments kann es sich um Parameter des gesamten Stoffstrommodells handeln, aber auch um Parameter von Submodellen. Wichtige Submodelle in Stoffstrommodellen sind die Prozesse. Auch diese können mit von außen zugänglichen Parametern versehen sein.

Zwecke der Parametrisierung

Mit der Parametrisierung im Rahmen des Life-Cycle Assessment verbinden sich folgende Vorteile:

- Formale Spezifikation des Einsatzbereichs: Der Einsatzbereich wird durch den Parameter charakterisiert. Außerhalb des zulässigen Wertebereichs für einen Parameter ist eine Prozessspezifikation nicht nutzbar. Aufzählungstypen für den Wertebereich ermöglichen die

Angabe von Technologien oder im Fall von Transportprozessen den Kfz-Typ.

- Flexibilisierung und Anpassung: Lebenszyklusdaten aus Datenbanken können besser an die Anforderungen der konkreten Stoffstromanalyse angepasst werden. Die Vorgänge können damit auch genauer abgebildet werden. Auch kann die Anzahl der Datensätze in der Datenbank dadurch reduziert werden.
- Experimente mit dem Modell: Ohne Parameter ist jedes Lebenszyklusmodell starr. Die Berechnung liefert genau ein Ergebnis. Will man mit dem Modell experimentieren, muss man normalerweise in die Tiefen eines Stoffstrommodells eingreifen. Die Parametrisierung ermöglicht hingegen kontrollierte Experimente; der Modellierer sieht systematisch Pfade für die Experimente vor. Verschiedene Szenarien entstehen in dem Fall durch verschiedene Parametersätze.
- Anschlussstellen für umrahmende Methoden wie Monte-Carlo-Simulation und Sensitivitätsanalysen: Das Lebenszyklusmodell wird „von außen“ systematisch durch die Parameter beeinflusst. Solche Methoden können auch als standardisierte Experimente mit dem Modell betrachtet werden.

Die Einführung von Parametern macht allerdings nur dann Sinn, wenn sie auf die Modellergebnisse einwirken. Im Zusammenhang mit dem Life-Cycle Assessment ist in dieser Hinsicht Vorsicht geboten, denn die Berechnung von Life-Cycle Inventories (LCI) erfordert zwingend die Verwendung von linearen Prozessspezifikationen. Dieser Umstand soll im Folgenden dargestellt werden. Danach wird es möglich sein, die Berücksichtigung und Verarbeitung als der eigentlichen LCI-Berechnung vorgelagerten Modellierungsschritt zu entwickeln.

Grundlagen des Life-Cycle Assessments

Beim Life-Cycle Assessment werden zwei verschiedene, formalisierte Modellierungsphasen miteinander kombiniert. In der ersten werden die anthropogenen Stoff- und Energieströme analysiert (Resultat: Life-Cycle Inventories), bei der zweiten die Wirkungen in der natürlichen Umwelt (Impact Assessment).

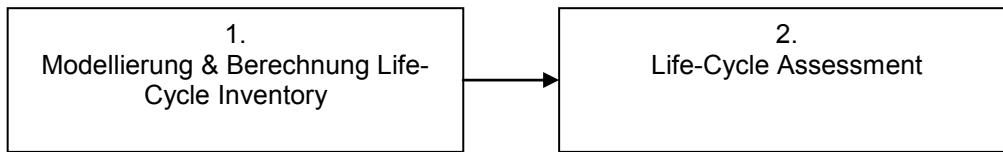


Abbildung 1. Die zwei formalisierten Modellierungsschritte des Life-Cycle Assessments

In den beiden formalisierten Modellierungsphasen wird streng dem Verursachungsprinzip gefolgt, wobei dieser Begriff bzw. der Begriff Ursache vieldeutig ist (vgl. Riebel 1990, S. 67). So kann das Verursachungsprinzip im kausalen Sinne interpretiert werden. Damit sind besondere Merkmale verbunden: „1. Die Ursache (als wirkender Zustand) muss der Wirkung (als bewirktem Zustand) zeitlich vorausgehen. 2. Die Ursache muss notwendige Voraussetzung für das Eintreffen der Wirkung sein, die nicht weggedacht werden kann, ohne dass die Wirkung mit Sicherheit entfiere (‘Keine Wirkung ohne Ursache’). 3. Wenn die Ursache vorliegt, muss zwangsläufig und ausnahmslos die Wirkung, und zwar stets dieselbe Wirkung eintreten. (‘Keine Ursache ohne Wirkung’. ‘Gleiche Ursache – gleiche Wirkung’)“ (Riebel 1990, S. 70). Das Verursachungsprinzip in der kausalen Interpretation ist von zentraler Bedeutung in den Naturwissenschaften. Aber auch Stoffstromanalysen sind von der kausalen Interpretation betroffen: Materie und Energie werden in Transformationsprozessen in andere Materie und Energie transformiert.

Das Verursachungsprinzip in der kausalen Interpretation ist nicht direkt anschlussfähig an den gedachten Nutzungskontext des Life-Cycle Assessments. Wie bereits skizziert, hat das Life-Cycle Assessment zum Ziel, umfassend die Umweltbelastungen zu bestimmen, die mit der Nutzung eines Produkts oder einer Dienstleistung verbunden sind. Oft geht es auch darum, zwei verschiedene Produkte oder Dienstleistungen zu vergleichen, die einen äquivalenten Nutzen stiften. Hier anschlussfähig ist eine andere Interpretation des Verursachungsprinzips: das Zweck-Mittel-Denken oder auch Finalprinzip: Für bestimmte Zwecke sollen bestimmte Mittel aufgewandt werden. Das Finalprinzip sieht also die hervorgebrachten Produkte und Dienstleistungen als die Zwecke an, derentwillen die Prozesse betrieben werden. Nach dem Finalprinzip sind demnach jeder positiven Wirkung als Zweck genau die negativen Wirkungen zuzurechnen, die ihretwillen in

Kauf genommen werden. Bildet man das Verhältnis aus Indikatoren für den Zweck und die Mittel, ergeben sich Effizienzkennzahlen. Aus dieser Nähe zwischen der Abschätzung negativer Wirkungen und Effizienzkennzahlen ergibt sich auch, dass das Life-Cycle Assessment bzw. die dem Life-Cycle Assessment zugrunde liegenden Methoden auch die Basis für wichtigen Effizienzkennzahlen für die Nachhaltigkeit sind: Ökoeffizienzkennzahlen, Materialeffizienzkennzahlen und Energieeffizienzkennzahlen. Allgemein kann eine Öko-Effizienzkennzahl definiert werden als der Quotient aus der quantitativen Beschreibung der Wertensetzung und einer Größe, welche die Umweltwirkungen in einer bestimmten Perspektive abbildet (vgl. Schaltegger, Burritt 2000, S. 50, 358), zum Beispiel Klimaveränderungen, Energieverbrauch oder Materialverbrauch.

Aus dem Finalprinzip leitet sich bereits grob ein Vorgehen für die Analyse ab: Zunächst wird die positive Wirkung bestimmt, auch quantitativ. Dann werden die mit der positiven Wirkung zusammenhängenden negativen Wirkungen bestimmt. Die Nähe zu den Vorgehensmodellen des Life-Cycle Assessments ist offensichtlich.

Bemerkenswert am Life-Cycle Assessment ist nun, dass zur Analyse der Zusammenhänge auf die Realprozesse verwiesen wird, und die Realprozesse folgen dem Verursachungsprinzip in der kausalen Interpretation (vgl. Riebel 1990, S. 71). Damit wird die Stoffstromanalyse, die dem Verursachungsprinzip in der kausalen Interpretation folgt, als Grundlage der Analyse betrachtet.

Mit anderen Worten: Dem Life-Cycle Assessment liegt das Verursachungsprinzip in zwei Interpretation zugrunde. Die Modellierung setzt nach dem Finalprinzip bei den einzelnen positiven Wirkungen an und leitet aus ihnen einen so genannten Referenzfluss ab. Diese werden dann auf der Stoff- und Energieflussebene dem Verursachungsprinzip in der kausalen Interpretation folgend der Lebenszyklusanalyse unterzogen und anschließend hinsichtlich der Umweltwirkungen verglichen. Die Folge davon ist, dass im Life-Cycle Assessment zwei verschiedene Interpretationen des Verursachungsprinzips auf spezifische Weise miteinander verschränkt sind. Diese Verschränkung hat zwei zentrale Auswirkungen auf die Modellierung: Erfassung des gesamten Lebenszyklusses zur umfassenden Beurteilung der Wirkungen in der natürlichen Umwelt, Anwendung von Allokationsregeln

bei Mehrproduktprozessen, keine Berücksichtigung von Beständen und Linearität der Prozessspezifikationen:

Ökologische Lebenszyklen

Ein zentrales Prinzip bei der Analyse ist die eindeutige Zurechenbarkeit. Zu erfassen sind beim Life-Cycle Assessment genau die Stoff- und Energieströme, die mit der Nutzenstiftung zusammenhängen: Zum einen dürfen, weil die Umweltwirkungen zu bestimmen sind, nicht zu wenige Stoff- und Energieströme erfasst werden und die Modellgrenzen nicht zu eng gezogen werden. Dies hängt mit dem zweiten Modellierungsschritt zusammen, in dem Aussagen über die Wirkungen der Stoff- und Energieströme in der natürlichen Umwelt außerhalb der Anthroposphäre gewonnen werden sollen (Wirkungsabschätzung, Impact Assessment). Die Wirkungsabschätzung ist unvollständig, wenn in der Stoffstromanalyse die Stoff- und Energieströme nicht bis zur Grenze zwischen Anthroposphäre und natürlicher Umwelt verfolgt werden. Es handelt sich dabei um eine Art Schnittstellenvereinbarung zwischen den beiden Schritten. Ein Grundsatz des Life-Cycle Assessments, das sich auch bereits in der englischen Bezeichnung der Methode manifestiert, ist, die Stoff- und Energieströme „von der Wiege bis zur Bahre“ zu modellieren.

Zurechnungen

Zum zweiten dürfen keinesfalls Umweltwirkungen in die Modellierung einbezogen werden, die nicht mit der Nutzung des Produkts bzw. Dienstleistung zusammenhängen. Dies führt zur so genannten Kuppelproduktionsproblematik. Viele Transformationen bringen gleich mehrere erwünschte Objekte hervor. Man denke nur an den Raffinerieprozess im Rahmen der Erdölverarbeitung. Typische Produkte sind Benzin, Diesel oder schweres Heizöl. Weil die Erdölverarbeitung heute in nahezu jeden Produktlebenszyklus einzubeziehen ist, hat man es mit zahlreichen Nebenprodukten zu tun, die in der Analyse nichts zu suchen haben. Es dürfen nur genau die Erdölprodukte einbezogen werden, die für die Herstellung, Nutzung und Entsorgung des betrachteten Produkts benötigt werden. Um dies im Stoffstrommodell zu gewährleisten, bedient man sich eines Tricks: Der Kuppelprozess wird in mehrere fiktive Einzelprozesse zerlegt. Jeder dieser fiktiven Einzelprozesse bringt genau ein erwünschtes Resultat hervor. Beispielsweise wird für den Transport von Getränkeverpackungen in der Re-

gel Diesel benötigt. Im produktbezogenen Stoffstrommodell des Life-Cycle Assessments wird der Dieselbedarf mit Hilfe eines fiktiven Dieselproduktionsprozesses gedeckt, der alle Schritte enthält, um Diesel herzustellen: Dieselförderung als Teil der Erdölförderung, Dieseltransport als Teil des Erdöltransports, Dieselraffinerieprozess und schließlich – nicht mehr mit der Kuppelproduktionsproblematik behaftet – der Dieseltransport zur Tankstelle.

Bestände

Eine weitere Auswirkung der Verknüpfung von Final- und Kausalprinzip bezieht sich auf die Modellierung und Berücksichtigung von Beständen im Life-Cycle Assessment. Aus dem Finalprinzip folgt, dass Bestandsveränderungen in der Modellierung keine Rolle spielen dürfen. Beispielsweise könnte es sein, dass für einen Verpackungsprozess zu einem bestimmten Zeitpunkt t_1 die Kartonverpackungen aus dem Warenausgangslager des Verpackungsmittelherstellers geliefert worden können und zu einem anderen Zeitpunkt t_2 müssen sie erst noch produziert werden. Von solchen Effekten an den Lagern ist im Life-Cycle Assessment zu abstrahieren, und die Anforderung, den gesamten Lebenszyklus zu berücksichtigen, verlangt an dieser Stelle, unabhängig von möglichen Lagerbestandsveränderungen die Produktion der Getränkeverpackung in die Analyse einzubeziehen.

Die Schlussfolgerung, die sich aus dieser Forderung ergibt, ist weit reichend: Im Life-Cycle Assessment werden keine Bestände modelliert. Für die Modellierung der Stoffstromsysteme ergibt sich damit, dass nur noch ein Typ von Systembestandteilen zu modellieren ist, dass also nur noch allein Transformationen bzw. Prozesse abgebildet werden und diese auch keine internen Bestände umfassen dürfen.

Für die Parametrisierung von LCI-Modellen bedeutet dies, dass alle Parameter, die sich auf Bestände beziehen, ebenfalls nicht einbezogen werden können. Dies kann zum Beispiel technologische Prozessparameter in der chemischen Industrie betreffen, welche sich auf Prozessgleichgewichte beziehen, die wiederum von Beständen und Substanzkonzentrationen abhängen.

Linearität

Darüber hinaus dürfen, dem Finalprinzip folgend, bei der Lebenszyklusanalyse Einflussgrößen keine Rolle spielen, die sich nicht auf der Basis des Finalprinzips aus dem Ziel der Analyse ableiten. Zum Beispiel wird man fordern, dass die quantitativen Ergebnisse der Analyse linear abhängig sind von der Höhe des Referenzflusses. Die Wahl der Höhe des Referenzflusses kann man daher von der Entscheidungssituation, etwa vom Alternativenvergleich, abhängig machen. So kann man die Analyse für einen Liter Getränk durchführen. Man darf daher davon ausgehen, dass man vor der Analyse den Referenzfluss mit einem Faktor 2 oder ähnlich versehen kann, nach der Analyse die Ergebnisse durch den Wert teilt und zu den gleichen Ergebnissen kommt.

Kern der LCI-Berechnung ist die Bestimmung der Prozessniveaus aller beteiligten Prozesse. Dafür werden in der Regel sequentielle Methoden oder Matrixmethoden verwendet. Die Prozessniveaus richten sich einzig und allein nach der Menge der vorgegebenen Referenzflüsse. Man kann sich in manchen LCA-Tools diese Prozessniveaus anzeigen lassen und dann auch die Input- und Outputstoffströme bestimmen. Diese Input/Output-Bilanzen haben aber nichts mit den realen Vorgängen einer Betrachtungsperiode zu tun. Sehr niedrige Werte für einen Prozess in einem LCI-Modell sagen also nicht aus, dass der Prozess auch in der Realität sehr niedrig ausgelastet gewesen ist. Folglich darf man auch nicht die evtl. festzustellende Tatsache in das Modell einfließen lassen, dass der betroffene Prozess bei niedriger Auslastung zu einem höheren Abfallaufkommen (Verschnitt o.ä.) führt.

Die Linearitätsanforderung hat direkten Einfluss auf die Abbildung des Realprozesses, etwa auf die Modellierung der Transformationen. Zulässig sind nur noch lineare Transformationen derart, dass mit Hilfe so genannter Produktionskoeffizienten ein lineares Verhältnis zwischen den verschiedenen Input- und Outputströmen angegeben werden muss; es ergibt sich ein Vektor aus Input- und Outputkoeffizienten. Nicht-lineare Beziehungen zwischen den Inputs und Outputs einer Transformation sind nicht zugelassen. Damit lassen sich beispielsweise die skizzierten Aspekte der Kapazitätsauslastung nicht direkt im Rahmen des Life-Cycle Assessments erfassen und abbilden.

Berechnungen zur Bestimmung von Life-Cycle Inventories

Die sich aus der Verknüpfung von Final- und Kausalprinzip ergebenden Anforderungen an das Life-Cycle Assessment führen direkt zum mathematischen Instrumentarium, das zur Beschreibung der produkt- bzw. dienstleistungsbezogenen anthropogenen Stoff- und Energieströme herangezogen werden kann und in ähnlicher Weise auch in der Kostenrechnung anzutreffen ist (vgl. Hejungs, Schmidt 1995, Möller 2000): die lineare Algebra. In einer Matrix wird die Vernetzung der verschiedenen beteiligten Transformationen abgebildet. Dabei kann auf die Vektoren, welche die Einzelprozesse beschreiben zurückgegriffen werden. Jede Zeile der Matrix steht für einen Stoff- oder Energiestrom, der zwischen den Transformationen ausgetauscht wird; alle Koeffizienten der Vektoren, die nicht am internen Stoff- und Energieaustausch beteiligt sind, bleiben hierbei unbeachtet. Ein lineares Gleichungssystem ergibt sich, wenn man auch noch eine rechte Seite ergänzt. Dieser Vektor ist sehr einfach aufgebaut: die Koeffizienten für die Zeile des Referenzflusses entsprechen der gewünschten Menge und alle anderen Koeffizienten sind gleich 0. Die Lösung des Gleichungssystems liefert dann die Prozessniveaus für die beteiligten Einzelprozesse. Nun können die Koeffizienten der Transformationen herangezogen werden, die nicht am internen Stoff- und Energieaustausch beteiligt sind und daher den Stoff- und Energieaustausch mit der Umwelt angeben. Die Prozessniveaus, multipliziert mit diesen Koeffizienten, liefern den prozessbezogenen Stoff- und Energieaustausch mit der Umwelt. Addiert man dann je Material alle Input- und Outputströme und nimmt den Referenzfluss hinzu, erhält man eine produktbezogene Input/Output-Darstellung. Auf der Inputseite wird angegeben, welche Stoff- und Energieströme in das anthropogene System auftreten, auf der Outputseite die, welche es verlassen. Eine Ausnahme bildet lediglich der Referenzfluss. Diese Darstellung bildet genau die Schnittstelle zur nachfolgenden Wirkungsabschätzung.

Schlussfolgerungen

Beim Life-Cycle Assessment werden also die Realprozesse in einer sehr speziellen Weise modelliert und in den Berechnungsalgorithmus eingespeist, die durch das Finalprinzip vorgegeben wird. Nicht-lineare Transformationen, Bestände sind in den Modellen nicht erlaubt, Zeitdynamik

nur aggregiert berücksichtigt usw. Für die Parametrisierung von Life-Cycle Assessments und die Lebenszyklusdaten ergeben sich folgende Folgerungen:

1.) Das Modellieren und die Parametrisierung von dynamischen Prozessen (vgl. Möller 2004), die der Abbildung dynamischer Vorgänge und der entsprechenden Einbeziehung von Beständen in das Modell dienen, ist im Rahmen des eigentlichen Life-Cycle Assessments nicht möglich und muss in ein vorgeschaltetes Simulationsmodell ausgelagert werden (vgl. Volz 1999, Möller 2005, Wohlgemuth 2005). LCA und Kostenrechnung sind hier nachgelagerte Auswertungsinstrumente zur Beurteilung der Wirtschaftlichkeit der modellierten dynamischen Systeme. Die Daten werden durch eine Protokollierung der dynamischen Prozesse gewonnen (vgl. Möller 2005).

2.) Üblicherweise sind statische Modelle die Grundlage des Life-Cycle Assessments; es wird angenommen, dass sich die dynamischen Prozesse sich dabei in einem eingeschwungenen Zustand befinden. Die Einbeziehung von Parametern, die sich auf einzelne Prozesse beziehen, muss dennoch vor der eigentlichen LCI-Berechnung erfolgen; es gibt keine Rückkopplung zwischen LCI-Berechnung und Parameterauswertung. Deswegen können die Ergebnisse der Parameterauswertung als Konstanten bzw. als unveränderliche Koeffizienten der linear spezifizierten Prozesse in die LCI-Berechnung einfließen. Anzumerken ist in dem Zusammenhang auch, dass die Umrechnung von Parametern in Prozesskoeffizienten durchaus mit Hilfe nicht-linearer Funktionen erfolgen kann. Die Linearitätsanforderung betrifft nur die eigentliche LCI-Berechnung.

3.) Ähnlich wie bei den Prozessparametern ist es denkbar, Modellparameter einzubeziehen und im Vorfelde in lineare Prozesskoeffizienten zu transformieren. Die Modellparameter können damit Einfluss auf mehrere Prozesse haben, während sich die Prozessparameter auf einen einzelnen Prozess beschränken. Die Modellparameter vereinfachen das Experimentieren mit Modellen. Software-Tools können Modellparameter mit geeigneten Elementen der Benutzungsoberfläche verknüpfen (Schieberegler, Drehknöpfe, Schalter usw.).

Formen der Parametrisierung

Im Folgenden soll beschrieben werden, wie die Einbeziehung der Parameter in Life-Cycle Assessments erfolgen kann. Diese Einbeziehung basiert auf einem vorgelagerten Modellierungs- und Berechnungsschritt: Die Transformation von Parameterwerten in lineare Prozesskoeffizienten, die dann in die LCI-Berechnung einfließen.

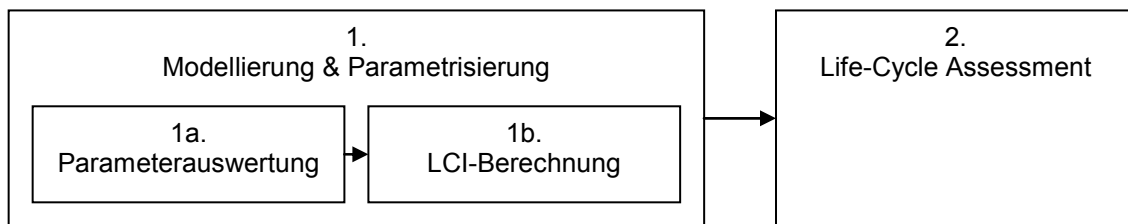


Abbildung 2. Dreistufiges Berechnungsverfahren für das parametrisierte Life-Cycle Assessment

Dabei kommen grundsätzlich zwei Verfahren in Betracht: Erstens die Berechnung eines periodenbezogenen Stoffstrommodells und zweitens die formelbasierte Parameterauswertung ohne umfassende vorgeschaltete Stoffstromanalyse.

Parametrisierung mit periodenbezogenen Stoffstrommodellen

Periodenbezogene Stoffstrommodelle werden insbesondere im betrieblichen Kontext erstellt. Wenn ein Unternehmen regelmäßig Stoffstromanalysen durchführt und die sich ergebenden Erkenntnisse systematisch in die Führungsprozesse einbezieht, kann von einem ökologischen Rechnungswesen gesprochen werden. Das ökologische Rechnungswesen basiert wie auch das externe und interne betriebliche Rechnungswesen auf einem bestimmten Rechnungsstil. Im Unterschied zum Life-Cycle Assessment werden nicht einzelne Produkte oder Dienstleistungen in den Blick genommen, vielmehr zielt das ökologische Rechnungswesen darauf ab, das gesamte Stoff- und Energieflussgeschehen in einer Organisation (Betriebsstandort, Unternehmen, Supply Chain) abzubilden und dann zweckgerichtete Auswertungen zu ermöglichen.

Mit dem Ansatz der Stoffstromnetze ist ein Rechnungsstil für Stoffstromanalysen und ökologische Rechnungswesen von Unternehmen entwickelt worden (vgl. Möller 2000). Es führt Konzepte des externen Rechnungswesens

sens (Doppik als Rechnungsstil der doppelten Buchführung) mit Konzepten aus der Informatik (Graphen) zusammen. Den Netzansatz aufgreifend bestehen die Stoffstromnetze aus zwei verschiedenen Typen von Knoten (Transitionen und Stellen) sowie Verbindungen zwischen ihnen. Transitionen dienen dazu, stoffliche und energetische Transformationen abzubilden, Stellen dazu, zeitliche Transformationen zu erfassen.

Dieses Konzept lässt es zu, die vernetzten Stoff- und Energieströme wie bei der doppelten Buchführung periodenweise aggregiert zu erfassen. Das Resultat sind periodenbezogene Stoffstrommodelle. Die einfachste Auswertung sind die Input/Output-Bilanzen, komplexere Auswertungen sind Life-Cycle Assessment und stoffstrombasierte Kostenrechnungen.

Hinsichtlich der Parametrisierung der Modelle und der Spezifikation der Prozesse sind die periodenbezogenen Stoffstromanalysen flexibler als das reine Life-Cycle Assessment: Nicht-lineare Prozessspezifikationen sind möglich; entsprechend einfach können Parameter eingeführt werden, sowohl auf Prozessebene (Transitionsparameter) als auch auf Modellebene (Netzparameter). Entsprechende Parameter sind in Software-Systemen (z.B. Umberto oder GaBi) implementiert worden.

Eine Transition modelliert in Stoffstromnetzen einen Prozess. Es sind eigenständige Submodelle, deren Beschreibung auf unterschiedliche Weise vorgenommen werden kann. Entsprechend können Parameter berücksichtigt werden

- Im einfachsten Fall werden die Inputs und Outputs des Prozesses über lineare Koeffizienten beschrieben. Eine Parametrisierung ist nicht möglich. Damit spielt dieser Fall für die Fragestellung der Parametrisierung von Datensätzen keine Rolle. Datensätze aus der EcoInvent Datenbank werden bspw. auf diese Weise abgebildet.
- Im Standard-Fall kommen Funktionen zum Einsatz (z.B. in Umberto User-Defined Functions, UDF genannt). Jeder Input und jeder Output bekommt einen Variablenbezeichner zugewiesen. Die Abhängigkeiten zwischen Inputs und Outputs können nun in nahezu beliebiger Weise vom Anwender über diese Funktionen definiert werden. Dabei stehen sowohl die Grundrechenarten, als auch Funktionen wie Wurzel, Logarithmus, Exponentialfunktion, Winkelfunktionen usw. zur Verfügung. Auf der Ebene jeder

Transition (Transitionsparameter) und jedes Netzes (Modell, Netzparameter) können vom Anwender Parameter definiert werden. Diese erhalten ebenfalls einen Bezeichner und einen Wert. Über ihren Bezeichner werden diese Parameter in die Funktionen einbezogen. Damit lassen sich die wichtigen Stellgrößen eines Prozesses aus den Funktionen herausziehen und für den Anwender komfortabel einstellbar machen. Dies ist für die Fragestellung Parametrisierung von Ökobilanz-Datensätzen der relevante Fall.

- Im erweiterten Fall werden die Prozesse mit Hilfe von Scriptsprachen beschrieben. Auch ist möglich, externe Komponenten für Submodelle einzubinden (COM-Modul oder .NET Assembly mit COM-Wrapper). Hier wird eine unbeschränkte Flexibilität geboten, den Prozess zu definieren und dabei auf interne Transitions- und Netzparameter sowie externe Parameter zuzugreifen. Dieser Fall ist aus Sicht der Bereitstellung von LCA-Datensätzen zu weit führend und wird nicht näher betrachtet.

Die zusätzliche Parametrisierung von linearen Prozessspezifikationen ist nicht erforderlich. Dies könnte sich ändern, wenn ein gemeinsamer Standard für parametrisierte Prozessspezifikationen entwickelt wird, der ohne vorgeschaltete periodenbezogene Stoffstromanalyse auskommen soll und damit auf die Parametrisierung linearer Prozessspezifikationen abzielt.

Die folgende Tabelle zeigt den Parametersatz für einen Lkw-Transport in Umberto(vgl. Mampel 1997, Schmidt et al. 1998):

Identifizier	Name	Value	Unit
C00	Entfernung (einfach)	200	km
C01	Auslastungsgrad (Hinfahrt)	60	%
C02	Auslastungsgrad (Rückfahrt)	40	%
C03	Fahrzeugtyp (1 bis 5, 6=Durchschnitt)	6	
C04	Fahranteil Autobahn	51	%
C05	Fahranteil Landstraße	30	%
C06	Fahranteil innerorts	19	%

Neben den Parametern können aber auch die Input- und Outputflüsse in die Berechnungen einbezogen werden. Die Input- und Outputflüsse sind entsprechend auch mit Identifikatoren versehen, so dass man auf ihre Werte zurückgreifen kann. Die Inputseite könnte folgenden Aufbau haben:

Identifizier	Name	Unit
X00	Tansportgut	kg
X01	Diesel	kg

Die Outputseite:

Identifizier	Name	Unit
Y00	Tansportgut	kg
Y01	Kohlendioxid, fossil	kg
Y02	Kohlenmonoxid	kg
Y03	NOx	kg
Y04	Schwefeldioxid	kg
Y05	NMVOC, unspez.	Kg
...

Bei Lkw-Transportprozessen wird üblicherweise vorausgesetzt, dass die Menge des Transportguts bekannt ist. Das heißt X00 oder Y00 sind bekannt und können in die Berechnung der unbekanntenen Werte X01, Y01, Y02, ... einbezogen werden. Der folgende Ausschnitt zeigt beispielhaft die Verwendung der bekannten Flüsse (X00, Y00) und der Parameter bei der Berechnung der Emissionen.

```

; Umrechnung der Auslastungsgrade zw. 0 und 1
LHIN = IF(>(C01,1),IF(<(C01,100),C01,100),1)/100
LZUR = IF(>(C02,0),IF(<(C02,100),C02,100),0)/100

; Berechnungshilfsgroessen fuer Anrechnung Hin/Rueckfahrt (siehe Beschreibung)
L00 = Lhin
LB = 2*Lhin/(Lhin+Lzur)
L01 = IF(=(C03,1),1,0)*C00*1.0E-5*X00/3750/Lhin
L02 = IF(=(C03,2),1,0)*C00*1.0E-5*X00/10500/Lhin
L03 = IF(=(C03,3),1,0)*C00*1.0E-5*X00/15300/Lhin
...
; Lkw-Mittel, gewichtet ueber alle Strassenkategorien
MKR6a= C04*(133.8*L00+240.3*LB)+C05*(93.3*L00+191.2*LB)+C06*(139.7*L00+228.7*LB)
NOX6a= C04*(4.597*L00+7.590*LB)+C05*(3.224*L00+5.760*LB)+C06*(4.616*L00+7.308*LB)
CO6a = C04*(0.464*L00+1.718*LB)+C05*(0.340*L00+1.667*LB)+C06*(1.188*L00+2.569*LB)
HC6a = C04*(0.098*L00+0.861*LB)+C05*(0.077*L00+0.882*LB)+C06*(0.790*L00+1.960*LB)
Par6a= C04*(0.122*L00+0.382*LB)+C05*(0.105*L00+0.332*LB)+C06*(0.249*L00+0.517*LB)
;
; Kraftstoffverbrauch und Emissionen insgesamt
MKRGa=(MKR1a*L01+MKR2a*L02+MKR3a*L03+MKR4a*L04+MKR5a*L05+MKR6a*L06)
X01 = MKRGa
Y01 = MKRGa*3.17533
Y02 = (CO1a*L01+CO2a*L02+CO3a*L03+CO4a*L04+CO5a*L05+CO6a*L06)
Y03 = (NOX1a*L01+NOX2a*L02+NOX3a*L03+NOX4a*L04+NOX5a*L05+NOX6a*L06)
Y04 = MKRGa*0.0009

```

Zu beachten ist, dass es sich um einen Ausschnitt handelt und Hilfsvariablen angelegt werden können. Davon wird im Datensatz intensiv Gebrauch gemacht.

Parametrisierung ohne periodenbezogene Stoffstromanalyse

Ohne vorgeschaltete Stoffstromanalyse werden die Parameter über – ggf. nicht-lineare Funktionen mit den linearen Koeffizienten verknüpft. Dabei können drei verschiedene Phasen der Einbindung von Parametern unterschieden werden:

- Die eigentlichen Parameter: Die Parameter sind mit Werten versehen. Sie sind innerhalb eines Berechnungsschritts unveränderlich. Man kann sie aber im Falle von Mehrfachrechnungen variieren (Monte-Carlo-Simulationen, Sensitivitätsanalysen, Parametervariation). Die Parameter sind mit einem Bezeichner (Identifikator) versehen, um sie in Ausdrücken verwenden zu können.
- Die Input- und Output-Prozesskoeffizienten: Diese bestehen für die Bestimmung der Koeffizienten, die Eingang in die LCI-Berechnungen finden, aus zwei Feldern: erstens für einen Faktor und zweitens für einen Identifikator (in GaBi Alias genannt). Hier können die Parameter direkt über den Identifikator einfließen.
- Funktionen: Funktionen ermöglichen die flexible Verknüpfung von Parametern mit den Input- und Output-Prozesskoeffizienten. Nicht-lineare und mehrstellige Funktionen können angegeben werden. Bei den Funktionen werden über die Identifikatoren Konstanten und andere Funktionen herangezogen. Ein Zugriff auf Input- und Outputflüsse ist dem Prinzip der Linearität folgend nur eingeschränkt möglich. (z.B. können in GaBi Flussgrößen wie der Heizwert oder Schwefelgehalt eines Flusses in die Berechnung eingehen).

Dieses Konzept der Einbeziehung von Parametern ermöglicht die Bestimmung der Prozesskoeffizienten „just in time“, wenn Parameter verändert werden: Die Veränderung eines Parameters bewirkt sofort die Neuberechnung aller Funktionen und Aliases, so dass sofort wieder eine lineare Prozessspezifikation für einen LCI-Berechnungslauf zur Verfügung steht. Das explizite Anstoßen einer Berechnung (wie bei

Umberto) ist nicht erforderlich. Die Tatsache, dass die Parameterauswertung ein vorgeschalteter Berechnungsschritt ist, muss für die Softwarenutzung nicht verstanden werden.

Volz und Florin verdeutlichen den Ansatz anhand eines Transportprozesses (Volz 1997, S. 75). Dabei sind Parametern entweder direkt ein Wert zugewiesen oder eine Formel zur Berechnung des Werts. Die Parameter Verbrauch und Emission werden im Beispiel über Funktionen berechnet:

Parameter	Formel	Wert
Distanz		200
Auslastung		1
Verbrauch	$0,0716 * \text{Auslastung}^{-0,929} * \text{Distanz}$	14,32
Emission	Verbrauch	14,32
Cargo		1

Diese Parameter können Prozessen und Plänen zugeordnet werden („process parameters“, „plan parameters“). Darüber hinaus sind auch globale Parameter in einer Datenbank möglich. Die Sichtbarkeit von Prozess- und Planparametern ist, dem einleitend beschriebenen Paradigma der funktionellen Programmierung folgend, auf einzelne Prozesse/Pläne und die aufrufende Ebene beschränkt (Pläne, in denen Prozesse/Unterpläne verwendet werden). So können die gleichen Bezeichner in verschiedenen Prozessen benutzt werden und jeweils eine andere Bedeutung haben.

In den Input- und Outputtabelle werden die Parameter über Aliase berücksichtigt. Der folgende Ausschnitt aus der Output-Tabelle zeigt die Verknüpfung von Alias und Faktor. Die Produktionskoeffizienten werden der besseren Anschauung wegen als „Menge“ ausgewiesen (z.B. $14,32 * 0,01596 = 0,22855$):

Alias	Fluss	Größe	Menge	Faktor
Cargo	Transportgut	Masse	1 kg	1
Emission	Kohlenmonoxid	Masse	0,22855 kg	0,01596
Emission	Kohlendioxid	Masse	44,75 kg	3,125
...

Im Unterschied zu typischen Spezifikationen von Transportprozessen in periodenbezogenen Stoffstromanalysen fließt hier die Transportmenge nicht über die In/Outputs in die Berechnung ein. Aufgrund des Linearitätsprinzips ist dies nicht möglich. Absoluten Transportmengen

gehen über den Parameter „Auslastung“ in die Berechnung ein, Verbräuche, Emissionen u. ä. werden dem Betrachtungsausschnitt (1kg Transportgut) proportional zugewiesen.

Eine ähnliche Lösung findet sich in SimaPro von Pre. Man kann Input-Parameter definieren, Funktionen spezifizieren – berechnete Parameter genannt – und die dann als Produktionskoeffizienten verwenden. Im Unterschied zu GaBi sind in SimaPro keine Alias-Felder in den Koeffiziententabellen vorgesehen, vielmehr können in den Koeffizientenspalten in den Wertefeldern per „Ausdruck“ Parameter mit Konstanten verknüpft werden. Man würde für einen Lkw-Transport in das Feld „Amount“ die Formel „Emission * 0.01596“ in der Zeile für Kohlenmonoxid eintragen.

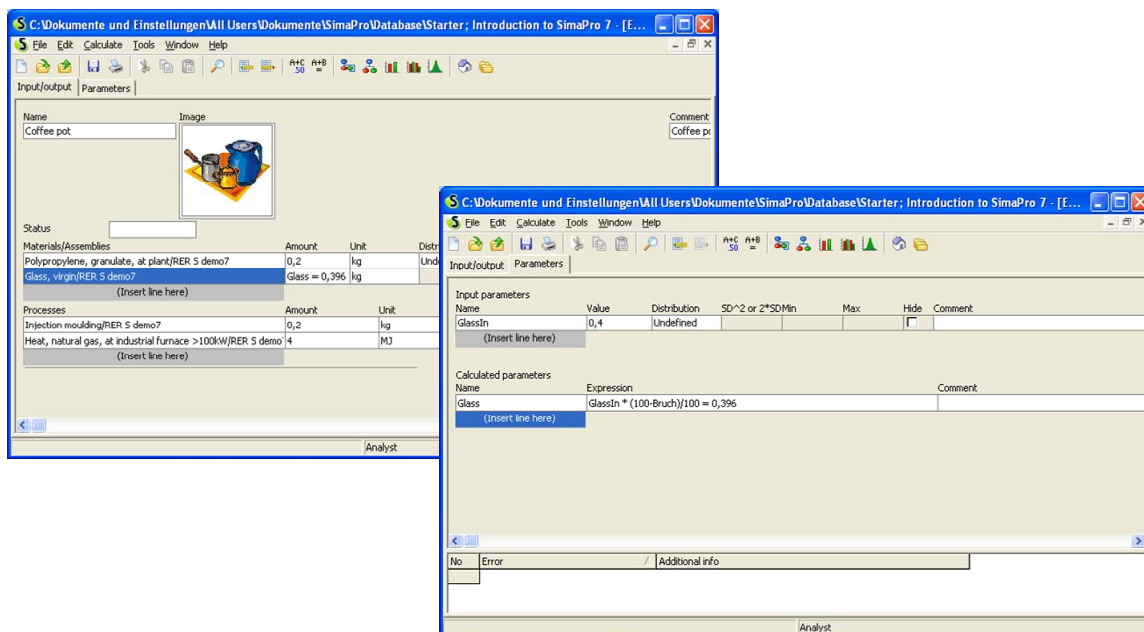


Abbildung 3. Parameter in SimaPro

In SimaPro ist für jeden Prozess bzw. Produktphase neben dem Register „Input / Output“ ein Register für Parameter vorgesehen. Hier können „Input Parameters“ und „calculated Parameters“ definiert werden. Bei den berechneten Parametern wird in Ausdrücken auf andere Parameter zugegriffen. Diese Parameter können sein:

- die für den Prozess angegebenen Inputparameter oder berechneten Parameter (soweit bereits berechnet),

- Inputparameter und berechnete Parameter auf Projektebene (projekt-globale Parameter),
- Inputparameter und berechnete Parameter auf Datenbankebene (datenbank-globale Parameter).

Auch SimaPro rechnet bei Änderungen der Parameter alle betroffenen Parameter und Koeffizienten „just in time“ durch. Das Ergebnis dieser Aktualisierungsrechnungen wird in den betroffenen auch sofort visualisiert. Nach dem Ausdruck wird, durch ein Gleichheitszeichen getrennt, das Berechnungsergebnis angezeigt. Erhält das Feld den Eingabefokus, dann verschwindet dieser Teil wieder, so dass der Ausdruck bearbeitet werden kann.

Im Ergebnis sind die Lösungen in GaBi und SimaPro sehr ähnlich. Es werden unterschieden:

- Inputparameter, die verändert werden können,
- Berechnete Parameter oder Funktionen, die mit Hilfe von Ausdrücken spezifiziert werden. In den Softwarewerkzeugen sind hierfür Formelinterpreter implementiert, welche der Formelauswertung dienen.

Die berechneten Parameter erlauben zwar die Verwendung von nicht-linearen Funktionen, allerdings beschränkt sich die Nicht-Linearität auf die Parameterauswertung und die Bestimmung der Produktionskoeffizienten. Dies wirkt sich auf die Eingangsgrößen dieser Berechnungen aus: Die dynamischen Elemente von Stoff- und Energieströmen sind keine Eingangsgrößen dieser Berechnungsschritte, sie können es auch nicht sein. Aufgrund der periodenbezogenen Stoffstromanalyse sieht das bei Umberto anders aus.

Datenformate für parametrisierte Lebenszyklusdaten

Die Lösungen von Umberto, GaBi und SimaPro unterscheiden sich aufgrund ihrer etwas unterschiedlichen Zielstellung. Umberto ist auf die Modellierung dynamischer Vorgänge ausgerichtet und lässt die Verknüpfung periodenbezogener Stoffstromanalysen mit Wirtschaftlichkeitsanalysen zu. GaBi und SimaPro zielen auf direkte und einfache Modellierungsweise und den Aufbau von wieder verwendbaren

Modulen für Datenbanken ab. Das Vorgehen, das in GaBi und SimaPro implementiert worden ist, kann als Standard angesehen werden. Umberto bietet bezüglich des Zeitaspekts weiter gehende Möglichkeiten.

Erweiterung vorhandener Datenformate

Im Folgenden sollen die Berücksichtigung am Beispiel des ELCD Formats (European Reference Life Cycle Data System) gezeigt werden. Dieses Format wurde durch die EU zum Zwecke der Harmonisierung der verschiedenen in Europa existierenden Daten(austausch)formate für LCA Informationen initiiert. Im ELCD Format sind Prozessparameter essentieller Bestandteil der Spezifikation, daher eignet sich dieses Format sehr gut zu Illustration der oben allgemein erläuterten LCA Parameter.

Eine Erweiterung um die Abbildung von Parameterinformationen lässt sich aber leicht auf andere Formate übertragen. Ein gemeinsamer Mindeststandard sollte folgende Merkmale aufweisen:

1.) Einfache Erweiterung der vorhandenen Datenformate für Life-Cycle Assessment um eine Parametrisierung. Wenn diese in einem Softwaretool nicht implementiert ist, sollte ein Datensatz auch ohne die Erweiterung mit Defaultparametern nutzbar sein. Im Falle eines XML-basierten Formats könnten alle Parametereinstellungen einfach ignoriert werden. Dies ist der Fall, wenn man in vorhandenen Daten für die Input- und Outputflüsse das bei Verwendung der default Parameter errechnete Ergebnis mit übergibt.

Im Falle des ELCD Datenformats bedeutet dies, bei <exchange>-Datensätzen die Elemente <resultingAmount>, <meanAmount> und <referenceToVariable> anzugeben:

- referenceToVariable: Dieses Element gibt den Bezeichner eines Parameters oder eines Funktionswerts an. Wenn dieses Element nicht angegeben leer ist, dann werden Parameter nicht berücksichtigt.
- meanAmount: Dieses Attribut den Faktor an, mit dem der Alias zu multiplizieren ist, um auf den Wert des resultingAmount - Elements zu kommen.

Ein Eintrag könnte im ELCD Format wie folgt aussehen:


```

<exchange dataSetInternalID="3">
  <referenceToFlowDataSet refObjectId="8864ce84-9967-11da-a72b-
0800200c9a66" version="01.00.001" type="flow data set"
  uri="../../flows/Carbon_dioxide_Emissions_to_air_8864ce84-9967-
11da-a72b-0800200c9a66_01.00.001.xml">
    ...
  </referenceToFlowDataSet>
  <exchangeDirection>Output</exchangeDirection>
  <referenceToVariable>Spec_CO2_wg</referenceToVariable>
  <meanAmount>0.001</meanAmount>
  <resultingAmount>0.00439505388554684</resultingAmount>
  <dataSourceType>Mixed primary / secondary</dataSourceType>
  <dataDerivationTypeStatus>Unknown or mixed derivation
  </dataDerivationTypeStatus>
</exchange>

```

Diese Spezifikationen können in GaBi direkt in die entsprechenden Felder eingetragen werden. Aus diesen Spezifikationen kann auch recht einfach ein Eintrag in einem „Amount“-Feld von SimaPro erzeugt werden, indem dort das Produkt aus Alias und meanAmounteingetragen wird.

2.) Bezogen auf den gesamten Datensatz müssen Parameter ergänzt werden. Die Parameterdatensätze sollten folgende Felder enthalten:

- Identifikator: Über den Identifikator kann der Parameter als Alias oder als Variable in einer Funktion verwendet werden. Dieses Feld darf nicht leer sein, es muss eindeutig innerhalb des gesamten Datensatzes sein, nur bestimmte Zeichen sind erlaubt, Groß- und Kleinschreibung spielt keine Rolle.
- Beschreibung für den Parameter (das Feld kann auch leer sein).
- Wert / Formel: Hier wird entweder ein Zahlenwert erwartet, der in den Funktionen und bei der Berechnung der Produktionskoeffizienten verwendet werden kann oder eine Formel, die die Berechnung des Parameterwertes beschreibt.
- Default Wert
- Minimalwert
- Maximalwert

3.) Um Abbildungen zwischen Parametern und Aliasen zu ermöglichen, können Funktionen angegeben werden. Die Funktionsdatensätze sollten folgende Attribute enthalten:

Identifikator: Über den Identifikator kann der Funktionswert als Alias oder als Variable in einer Funktion verwendet werden. Wie bei den Parametern gelten auch hier die folgenden Einschränkungen: Dieses Feld darf nicht leer sein, es muss eindeutig innerhalb des gesamten Datensatzes sein, nur bestimmte Zeichen sind erlaubt, Groß- und Kleinschreibung spielt keine.

Beschreibung für die Funktion (das Feld kann auch leer sein).

Ausdruck: Hier wird in Form einer Zeichenkette ein mathematischer Ausdruck angegeben. Die Identifikatoren der Parameter und anderer Funktionen können im Ausdruck verwendet werden. Diese werden über eine Reihe von vordefinierten Funktionen miteinander verknüpft. Neben den Grundrechenarten sind auch höhere Funktionen wie Wurzel, Quadrat, Sinus, Cosinus usw. vorzusehen. Die Formelsprache muss in einer Grammatik (ergänzt um eine Beschreibung der Semantik) festgelegt werden, damit der Datenaustausch zwischen verschiedenen Softwaretools funktioniert; ansonsten scheint die Datenübernahme an unterschiedlich implementierten Formelinterpretern.

Zur Illustration soll am Beispiel des ELCD-Formats folgendes Beispiel dienen:

```
<mathematicalRelations>
  <modelDescription xml:lang="en">The emission profile as well as the fuel
    consumption is calculated according to the settings of the variable
    parameters as described in the technology description. The calculation
    formulas of the single variables are given in the section variable /
    parameter.</modelDescription>
  <variableParameter variableParameter="Distance" meanValue="100"
    minimumValue="1" maximumValue="10000" comment="[km] distance start -
    end, default = 100 km" />
  <variableParameter variableParameter="Payload" meanValue="27"
    formula="27" comment="[t] default = 27 t" />
  <variableParameter variableParameter="ppm_Sulfur" meanValue="50"
    minimumValue="0" maximumValue="500" comment="[ppm] sulphur content in
    diesel, default Europe since 2003 = 50 ppm" />
```

```

<variableParameter variableParameter="Share_Check" meanValue="1"
  formula="Share_MW+Share_IU+Share_UR" comment="Check - value must be 1"
  />
<variableParameter variableParameter="Share_IU" meanValue="0.24"
  minimumValue="0" maximumValue="1" comment="[-] driving share
  interurban, default = 0.43" />
<variableParameter variableParameter="Share_MW" meanValue="0.68"
  minimumValue="0" maximumValue="1" comment="[-] driving share motorway,
  default = 0.27" />
<variableParameter variableParameter="Share_UR" meanValue="0.08"
  minimumValue="0" maximumValue="1" comment="[-] driving share urban,
  default = 0.30" />
<variableParameter variableParameter="Spec_Benzene_IU"
  meanValue="2.12329758169935E-7" formula="(0.004798944+(0.004886031-
  0.004798944)*Utilisation)/(Payload*1000*Utilisation)"
  comment="[gbenzene/(kg*km)] benzene emission interurban" />
<variableParameter variableParameter="Spec_Benzene_MW"
  meanValue="2.31094651416122E-7" formula="(0.005558006+(0.005258731-
  0.005558006)*Utilisation)/(Payload*1000*Utilisation)"
  comment="[gbenzene/(kg*km)] benzene emission motorway" />
<variableParameter variableParameter="Spec_Benzene_UR"
  meanValue="5.65771616557734E-7" formula="(0.010704031+(0.013386887-
  0.010704031)*Utilisation)/(Payload*1000*Utilisation)"
  comment="[gbenzene/(kg*km)] benzene emission urban" />
<variableParameter variableParameter="Spec_Benzene_wg"
  meanValue="2.53365234248366E-5"
  formula="( (Share_MW*Spec_Benzene_MW)+(Share_IU*Spec_Benzene_IU)+(Share
  _UR*Spec_Benzene_UR) ) *Distance" comment="[gbenzene/kg] benzene
  emission weighted" />
</mathematicalRelations>

```

Zu beachten ist, wenn man sich die Zahlenwerte im Beispieldatensatz ansieht, dass die Zahlenwerte nach amerikanischem Standard als Dezimaltrennzeichen den Punkt „.“ verwenden. Dies kann man gegenüber dem Modellierer bzw. Benutzer verbergen, indem man in Eingabe- und Ausgabefeldern eine Anpassung an nationale Standards vornimmt. Entwicklungsumgebungen und Betriebssysteme unterstützen dies, so dass hiermit nicht vorgeschrieben wird, auf welche Weise die Daten eingegeben und ausgewiesen werden.

Grammatik für Funktionsausdrücke

Damit die Funktionen über die Grenzen einzelner Softwaretools hinweg einheitlich „verstanden“ und interpretiert werden, soll im Folgenden ein Standard für die Funktionsausdrücke vorgeschlagen werden, der eine Reihe vordefinierter Funktionen vorsieht (sqrt, sqr, exp usw.). Daneben werden aber auch eine Reihe boolescher Funktionen vorgesehen; diese sind insbesondere im Zusammenhang mit bedingten Ausdrücken if(exp1;

exp2; exp3) von Bedeutung, so dass Fallunterscheidungen unterstützt werden.

Die folgende Grammatik ist in EBNF formuliert. Die Nonterminale sind in eckigen Klammern dargestellt, die Terminale in Anführungszeichen. Eine Reihe von Funktionen ist doppelt vorhanden. Dies wird vorgeschlagen, um eher eine Obermenge zu erzeugen, so dass vorhandene Ausdrücke in den Software-Tools weiter funktionieren und die vorhandenen Interpreter lediglich erweitert werden müssen (Vermeidung von Inkompatibilitäten).

```
expression      = factor { ( "+" | "-" ) factor };
factor          = pwrExpr { ( "*" | "/" | "div" | "mod" ) pwrExpr };
pwrExpr        = simpleExpr { "^" simpleExpr };
simpleExpr      = number | numConst | identifier | function |
                "(" expression ")";
function       = function1 | function2 | boolFunction;
function1      = ( "sqr" | "sqrt" | "round" | "exp" | "ln" | "lg" |
                "abs" | "ceil" | "int" | "trunc" | "frac" | "floor" |
                "ipower" |
                "tan" | "cos" | "sin" | "arctan" | "arccos" |
                "arcsin" | "cotan" | "sinh" | "cosh" | "tanh" |
                "asin" | "acos" | "atan" ) "(" expression ")";
function2      = ("power" | "min" | "max")
                "(" expression listSep expression ")";
boolFunction   = ( "iif" | "if" )
                "(" boolExpr listSep expression listSep expression ")";
boolExpr       = ( boolConst | comparison )
                { boolOperator ( boolConst | comparison ) };
comparison     = ( expression compareOperator expression ) |
                "(" comparison ")";
boolOperator   = "&" | "|" | "and" | "or" | "xor";
compareOperator = "==" | "<" | ">" | "!=" | "=" | "<>" | "<=" | ">=";
identifier     = (letter | "_" ) {character};
number        = ["-"] digit {digit} [ decSep digit {digit} ]
                [ ["E" | "e" ] ["-"] digit {digit} ];
letter        = "A" | "B" ... | "Z" | "a" | "b" | ... | "z";
digit         = "0" | "1" | ... | "8" | "9";
character     = letter | digit | "_";
boolConst     = "true" | "false";
numConst      = "pi";
decSep        = ".";
listSep       = ";";
```

Im Folgenden soll die Semantik beschrieben werden:

- „==“ und „=“ haben die gleiche Interpretation (keine Unterscheid wie in manchen Programmiersprachen hinsichtlich Gleichheit und Identität)

- „!=“ und „<>“ haben die gleiche Interpretation
- „&“ und „and“ haben die gleiche Interpretation
- „|“ und „or“ haben die gleiche Interpretation
- Gleiche Interpretation: „arcsin“ = „asin“, „arctan“ = „atan“, „arccos“ = „acos“
- „ipower(x,y)“ verwendet als y eine Integerzahl
- „div“ und „mod“ dienen der ganzzahligen Division bzw. modulo (wie in der Programmiersprache Pascal, Problem: Es gibt auch Interpreter, die „mod“ als Funktion mit 2 Parametern vorsehen.
- „^“ und „power“ haben die gleiche Bedeutung und unterscheiden sich durch die Infix- und Prefixschreibweise.
- Die Funktionen „if“ und „iif“ haben die gleiche Bedeutung. Der if-Funktion kommt eine Wächterfunktion zu: Wenn das Ergebnis des booleschen Ausdruck true (also ungleich Null) ist, dann wird nur der zweite Parameter ausgewertet, ansonsten nur der dritte. Beispielsweise kann man im booleschen Ausdruck auf Null prüfen, um eine Division durch Null zu vermeiden, also zum Beispiel `if(x==0; 1; 1/x)`.
- Bei den Ausdrücken muss wie auch bei den Zahlenwerten festgelegt werden, welches Dezimaltrennzeichen zu verwenden ist. Dieser Vorschlag orientiert sich an den Standards für Programmiersprachen und verwendet den Punkt „.“ als Dezimaltrennzeichen. Dies ist erforderlich, um die Datensätze international austauschbar zu machen. Ähnlich wie bei Zahlenwerten können die Ausdrücke an nationale Festlegungen angepasst werden, indem der Punkt bei der Eingabe und Ausgabe durch das jeweils gültige Trennzeichen ersetzt wird. Dies allerdings geht nur, wenn das Dezimaltrennzeichen in der Grammatik nicht an anderer Stelle verwendet wird. Dies ist insbesondere bei Parameterlisten mehrstelliger Funktionen der Fall. Weil das Komma oft als Trennzeichen verwendet wird, wird hier vorgeschlagen, das Semikolon in Parameterlisten zu verwenden. Um

Mehrdeutigkeit zu vermeiden, ist zusätzlich festzulegen, dass, wenn das Semikolon als nationales Dezimaltrennzeichen festgelegt wird, dies dennoch im Interpreter nicht gilt; dann ist der Punkt als Trennzeichen zu verwenden.

Erweiterungen

Die Parametrisierung erlaubt wesentlichen Verbesserungen bei der Erstellung von Lebenszyklusdaten. Dazu sollen 2 Beispiele vorgestellt werden, die man eventuell in entsprechende Standards aufnehmen kann. Zuvor sollte jedoch die Praxisrelevanz der Erweiterungen geprüft werden.

Selbstdefinierte Funktionen

Die Grammatik des Formelinterpreters sieht eine Reihe fest vorgegebener Funktionen wie `sin`, `exp` oder `round` vor. Bestimmte Klassen von Funktionen lassen sich einfach und komfortabel durch den Modellierungsexperten spezifizieren. Drei solche Klassen sollen im Folgenden vorgeschlagen werden: Stückweise lineare Interpolation, Treppenfunktion und lineare Regression.

Die folgende Abbildung zeigt die stückweise lineare Interpolation. Diese Funktion dient dazu, vom Auslastungsgrad auf den Verschnitt zu schließen. Die Funktion basiert auf einer Liste von Stützstellen, die für die Maschine erhoben worden sind. Sie kann nach und nach durch weitere Messwerte bzw. Stützstellen verfeinert werden.

Für die Einbindung in die Parameterauswertung ergeben sich zwei verschiedene Ansätze:

- Man erweitert den Funktionssatz des Formelinterpreters. Dieser sich an funktionalen Programmiersprachen orientierende Ansatz hat den Vorteil, dass man verschiedene Eingangsparameter vorsehen kann.
- Man betrachtet die selbst-definierte Funktion als eine Zeile in der Liste der Funktionen. Man gibt hier nicht mehr einen Ausdruck an. Vielmehr wird der Ausdruck durch die graphisch visualisierte stückweise lineare Interpolation ersetzt (eine ähnliche Lösung hat

man beispielsweise beim System-Dynamics-Tool Stella gewählt (vgl. Hannon, Ruth 2001, S. 18).

Die folgende Abbildung geht von der zweiten Lösung aus. Im Falle der Aufnahme in einen Parametrisierungsstandard ist zu beachten, dass die zweite Variante deutlich einfacher zu implementieren ist.

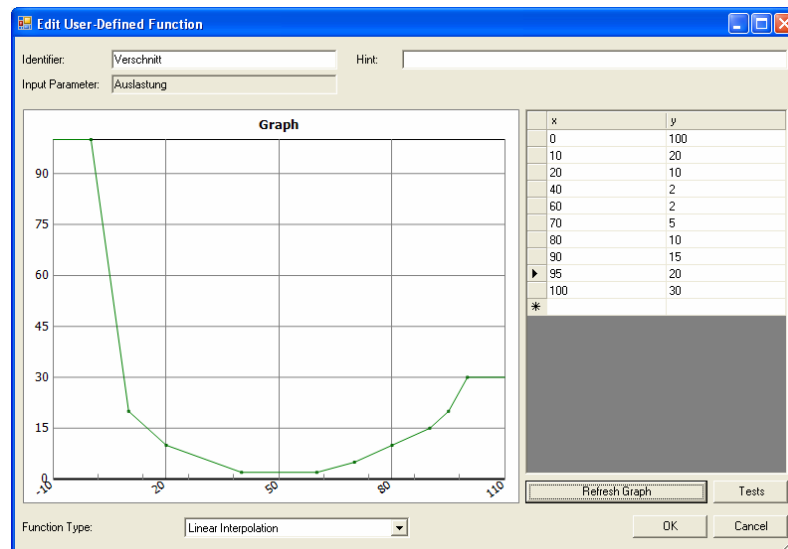


Abbildung 4. Stückweise lineare Interpolation für das Verhältnis zwischen Auslastungsgrad und Verschnitt einer Maschine

Eine Abwandlung der stückweise linearen Interpolation ist die Treppenfunktion. Die Wertetabelle dient hier dazu, die Sprungstellen zu definieren. Eine solche Funktionsklasse kann beispielsweise dazu dienen, die Technologie in Abhängigkeit von der Auslastung festzulegen (zur Abbildung verschiedener Technologien siehe die zweite Erweiterung unten).

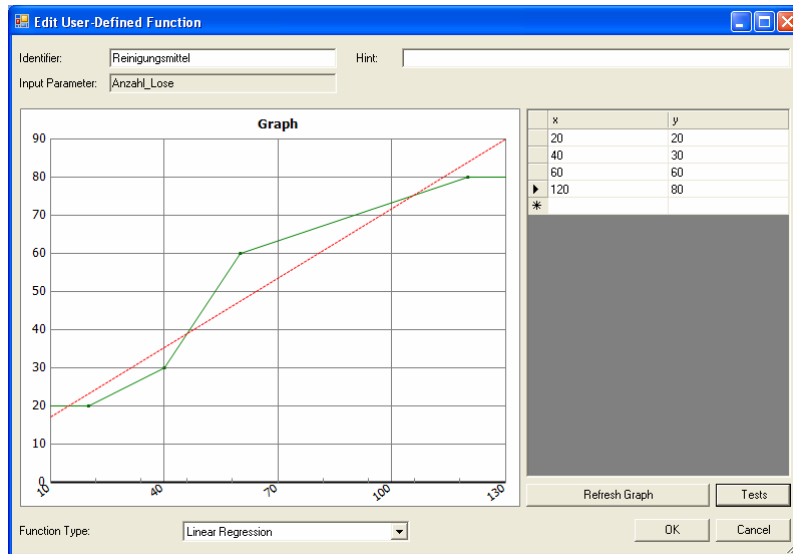


Abbildung 5. Verwendung der linearen Regression zur Bestimmung der Menge eines Reinigungsmittels in Abhängigkeit von der Anzahl unterschiedlicher Produktionslose

Ebenfalls mit Stützstellen bzw. Messwerten arbeitet die lineare Regression. Die lineare Regression dürfte eines der wichtigsten Instrumente zur Bestimmung von Prozessspezifikationen sein.

Die benutzerdefinierten Funktionen sind durch die Stützstellen und die Funktionsklasse eindeutig spezifiziert. Entsprechend einfach lassen sich die Datenstrukturen erweitern. Der folgende Ausschnitt aus einer XML-Datei zeigt die Spezifikation der linearen Regression aus der obigen Abbildung:

```
<functions>
  <function
    identifier="Reinigungsmittel"
    description=""
    inputParameter="Anzahl_Lose"
    type="LinearRegression"
    unit="">
    <nodes>
      <node x="20" y="20"/>
      <node x="40" y="30"/>
      <node x="60" y="60"/>
      <node x="120" y="80"/>
    </nodes>
  </function>
</functions>
```



```
</function>  
</functions>
```

Die Erweiterung in den Datenformaten ist damit einfach und konsistent. Die Funktionen sind so genau spezifiziert, dass ein unterschiedliches Verhalten verschiedener Software-Tools nicht zu erwarten ist.

Kreuztabellen

Die Erweiterungen zur Parametrisierung gehen davon aus, dass über die Parameter wichtige Kenngrößen eines Prozesses berechnet werden können. Diese Kenngrößen charakterisieren den gesamten Prozess. Allerdings handelt es sich um skalare Größen. Sie sind nicht dafür vorgesehen, ganze Produktionskoeffizientensätze in einem Vektor zusammenzufassen.

Die Lösung bei Lebenszyklusdaten besteht darin, für jeden in Betracht kommenden Vektor einen Datensatz anzulegen. Zum Beispiel legt man für jeden Lkw-Typ einen eigenständigen Datensatz an. Auch bei Produktionstechnologien muss man für verschiedene Technologien mehrere Datensätze anlegen. Damit fällt eine ganze Kategorie von Parametern faktisch weg.

Um aus Produktionskoeffizientenvektoren parameterabhängig zu selektieren zu können, werden Kreuztabellen vorgeschlagen. Die Produktionskoeffizientensätze werden als Spaltenvektoren erfasst. Ein ganzzahliger Eingangsparameter wählt den zu verwendenden Vektor aus. Eine weitere Eingangsgröße ermöglicht die Berücksichtigung von Kenngrößen.

Das folgende Beispiel zeigt, wie das Beispiel aus Volz und Florin (1997, S. 75) um eine Kreuztabelle für den Lkw-Typ erweitert wird:

Alias	Typ = 0	Typ = 1	Typ = 2	Typ = 3
CO	0.01596	0.01389	0.01338	0.0101
CO2	3.125	3.025	3.006	2.985
NOx	0.028241	0.02067	0.0270	0.0209
...

Eingangsparameter dieser Kreuztabelle sind der Lkw-Typ „Typ“ und, um nahe am Beispiel zu bleiben, der Alias „Emission“. So wird der Alias „CO“ beim Lkw-Typ 0 wie folgt berechnet: $CO = 0.01596 * Emission = 0.01596 * 14.32$, wobei $Emission = 14.32$ wie im Beispiel angenommen wird. Gibt man als Lkw-Typ stattdessen 1 vor, dann ergibt sich $CO = 0.01389 * Emission = 0.01389 * 14.32$.

Die Tabellen für die sich ergebenden Produktionskoeffizienten werden entsprechend angepasst (bei den Zahlenwerten ist der Lkw-Typ 0 gewählt worden):

Alias	Fluss	Größe	Menge	Faktor
Cargo	Transportgut	Masse	1 kg	1
CO	Kohlenmonoxid	Masse	0,22855 kg	1
CO2	Kohlendioxid	Masse	44,75 kg	1
NOx	Stickoxide	Masse	0,40441 kg	1
...

Auch die Kreuztabellen lassen sich einfach in die Datenstrukturen integrieren. Wie auch die `<functions>` sind die Kreuztabelle Unterknoten der Struktur `<parameters>`:

```
<crossTabs>
  <crossTab
    description=""
    inputIndicator="Emission"
    rowSelector="Typ"
    colCount="4">
    <rows>
      <row Identifier="CO">
        <item index="0" value="0.01596" />
        <item index="1" value="0.01389" />
        [...]
      </row>
      <row Identifier="CO2">
        <item index="0" value="3.125" />
        <item index="1" value="3.025" />
        [...]
      </row>
    </rows>
  </crossTab>
</crossTabs>
```

Es bietet sich an, die Benutzungsoberfläche für die Erfassung und Bearbeitung von Kreuztabellen in Anlehnung an Tabellenkalkulationsprogramme zu gestalten.

Resümee

Die Parametrisierung von Lebenszyklusdaten sowie die Parametrisierung der Stoffstrommodelle bringt eine Reihe von Vorteilen mit sich. Bezogen auf die Datensätze in Prozessdatenbanken sind es insbesondere zwei Vorteile:

- Die Datensätze können flexibler modelliert werden und lassen sich besser an unterschiedliche Einsatzkontexte anpassen.
- Die Parametrisierung bietet einen systematischen Weg der Charakterisierung und der Eingrenzung des Einsatzkontextes, der sich durch den Wertebereich der Parameter spezifizieren lässt.

Aber auch für die Stoffstromanalysen verbindet sich mit der Parametrisierung eine Reihe von Vorteilen:

- Die Modelle werden flexibler und ermöglichen systematische Experimente mit Modellen.
- Insbesondere sind sie die Anschlussstellen für Monte-Carlo-Simulationen und Sensitivitätsanalysen. Auch eine „manuelle“ Optimierung der Modelle ist möglich, ohne dass in die Tiefen eines Modells eingegriffen werden muss.

Die Berücksichtigung von Parametern für das Life-Cycle Assessment in den Datensätzen erweist sich als vergleichsweise einfach. Vorhandene Datenformate müssen nur um wenige Strukturen erweitert werden. Bei auf der Auszeichnungssprache XML basierenden Formaten kann diese Erweiterung so vorgenommen werden, dass auch Softwaretools diese Datensätze lesen können, die den Umgang mit Parametern nicht beherrschen (diese arbeiten dann mit den Default-Werten).

Besondere Aufmerksamkeit erfordern bestimmte Details, die bei Nicht-Beachtung zu erheblichen Schwierigkeiten führen. Das betrifft das Zahlenformat und insbesondere die Interpretation von Funktionsausdrücken.

Literatur

- Hannon, B., Ruth, M. (2001): Dynamic Modeling, Second Ed., Berlin, Heidelberg, New York
- Heijungs, R. (1994): A generic method for the identification of options for cleaner production. In: Ecological Economics 10, s. 69-81
- Mampel, U. (1997): Die Prozeßbibliothek in Umberto. In: Schmidt, M., Häuslein, A. (Hrsg.): Ökobilanzierung mit Computerunterstützung, Berlin, Heidelberg, New York
- Möller, A. (2000): Grundlagen stoffstrombasierter Betrieblicher Umweltinformationssysteme, Bochum, Germany
- Möller, A. (2004): Continuous Simulation in Material Flow Networks. In: Proceedings of the iEMSs 2004 (International Environmental Modelling and Software Society) "Complexity and Integrated Resource Management", Osnabrueck, Germany
- Möller, A. (2005): Dynamic Material Flow Analysis in the Life Cycle Assessment Tool Chain. In: Geldermann, J., Treitz, M., Schollenberger, H., Rentz, O. (Eds.): Challenges for Industrial Production, Karlsruhe, Germany
- Schmidt, M. (1995): Die Modellierung von Stoffrekursionen in Ökobilanzen. In: Schmidt, M., Schorb, A. (Hrsg.): Stoffstromanalysen in Ökobilanzen und Öko-Audits, Berlin, Heidelberg, New York
- Schmidt, M. et al. (1998): Evaluierung gängiger Datenmodelle zur Ermittlung verkehrlicher Umweltbelastungen. In: Haasis, H.D., Ranze, K.C. (Hrsg.): Umweltinformatik 98, Band 1, Marburg, Germany
- Volz, T., Florin, H. (1997): Computereinsatz bei der Ganzheitlichen Bilanzierung. In: Geiger, W. et al. (Hrsg.): Umweltinformatik '97, Band 1, Marburg, Germany
- Volz, T. (1999): Integration systematischer Analyse und Prognose in die Ganzheitliche Bilanzierung – Instrumentarium zur rechnergestützten Modellierung. Stuttgart, Universität, Fakultät 13, Institut für Kunststoffkunde und Kunststoffprüfung, Dissertation, 1999
- Wohlgemuth, V. (2005): Komponentenbasierte Unterstützung von Methoden der Modellbildung und Simulation im Einsatzkontext des betrieblichen Umweltschutzes, Aachen, Germany