General requirements for LCA software tools

N. Unger, P. Beigl and G. Wassermann

Institute of Waste Management, BOKU – University of Natural Resources and Applied Life Sciences, Vienna, Austria

Abstract: In recent years Life Cycle Assessment (LCA) software tools have become increasingly important. Today a large number of LCA programs are available. The large diversity of LCA software tools on offer makes it necessary to first pinpoint some general requirements that determine the quality of an LCA software tool and secondly to describe these requirements qualitatively. The authors last year assessed four different software tools by using each of these programs to model waste management scenarios . This modelling process revealed many differences in the quality of the LCA software tools. These differences are unrelated to the kind of modelled scenario and therefore are relevant for all kinds of LCAs (e.g. product LCA). Using these experiences and by conducting additional research in the literature, we have deduced some general requirements considered essential for good quality LCA software tools. These will be presented in this paper.

Key words: LCA, software, features, quality

1 INTRODUCTION

Life Cycle Assessment (LCA) is today an often used method for assessing the potential environmental impact of products, services or proceedings. Software tools were developed to make the processing and calculation of LCAs easier. The first steps were taken about two decades ago, with the main focus often on the assessment of production processes. Over time LCA software was also applied to other fields such as waste management.

There is a large variety of LCA software tools on the market. The foremost – and for the potential user also often prohibitive – property of a software tool is the price. The price of an LCA software tool can vary between several thousand euros and free of charge. Some tools offer a wider range of features than others. Some are focussed on a specific field of LCA, e.g. LCA in waste management, while others try to cover different application fields of LCA. Also the data and data quality can have an effect on the price of a software tool.

Depending on the purpose for which the user has selected the software, different LCA tools are more suitable for particular applications. However, a number of properties and features are essential for any good quality LCA software tool regardless of the kind of user and the kind of LCA it is being used for. This paper will discuss which features are important and which requirements are desirable for a good LCA application.

The content of this paper is mostly gleaned from experiences with four software tools for LCAs in waste management, as well as from test versions of other LCA software tools.

2 WHO USES LCA SOFTWARE TOOLS?

Different groups of LCA software users can be distinguished. The first group includes scientists and researchers. Users in this group are often experienced with LCA and have a good knowledge and understanding of the context and the features of the LCA method. Thus they make high demands on LCA software tools: They need a flexible software tool that enables them to model "common" often-modelled scenarios as well as scenarios that diverge from the standard. Also the tool should support modelling of complex process chains. The provided data need to be of good quality (see 4.3) and adequate, particularly because, in contrast to business users, scientists usually do not have their own data. It should be possible to create new data sets. In addition, scientists need the freedom to make their own improvements and modifications to existing data, specifications and parameters.

Industry, on the other hand, uses LCA software to improve its environmental performance, for process optimisation and product development. The users want “ready-to-use” software, where many of the specifications are already pre-set with only a few parameters needing to be determined.

Also decision makers use LCA to compare different solution options and hence also LCA software tools. Decision makers generally want an easy-to-understand presentation of the results in terms of which option is the best.

The developers of LCA tools aim to serve both groups of users: scientists and practical users from industry. It is very expensive to develop a software tool and thus it can only pay off when it is sold to the widest possible audience [Rizzoli and Young, 1997].

Not all of the mentioned requirements need to be fulfilled by a software tool in order to be acceptable to a specific user group.

3 WHY ARE LCA SOFTWARE TOOLS USED?

Environmental processes are often very complex and convoluted. This makes it difficult to model an LCA. Additionally LCA is often data intensive. Computers and adequate software tools are thus used to support the user in managing and editing these amounts of data. LCA software further helps to structure the modelled scenario, displaying the process chains and presenting and analysing the results. LCA software tools can be used whenever the method of LCA is applied.

The main reason for using LCA is to calculate the environmental aspects and potential impact associated with a product (ISO 14040). Also environmental hot spots (processes that have a large impact on the environment) can be identified. A more environmentally-friendly production process can thus be developed where they are most effective. LCA can also be used for a cleaner approach to production. It can help to improve and optimise resource management, which leads to a more efficient use of materials and energy.

LCA therefore is used mainly for comparing different options and for deciding which option is best for the environment. LCA and LCA software are thus used as a support tool in decision taking.

4 EXAMPLES OF TECHNICAL AND METHODOLOGICAL REQUIREMENTS

People who wish to use an LCA software tool often face the dilemma of which tool is best for their purposes. There are some software comparisons available that can help (cf. Jönbrink et al., 2000; Frühbrodt, 2002; Unger, 2003). An overview of some properties and features of commercial LCA software tools is provided here. Additional desirable features are pointed out. These can be seen as general requirements that need to be fulfilled by a good LCA software tool.

4.1 Structure and display of processes

A software tool generally consists of a database and a modelling module. The data are handled and modelled on an interface.

The modelling consists mainly of connecting successive processes with material flows. They build the process chain. Each process represents a stage in production and is defined by its input and output (see 4.4). The output from a preceding process builds the input for the next process. Simple process chains can be modelled in one layer. To handle more complex process chains, a hierarchical structure, as displayed in Figure 1, is needed. The main process stages, e.g. extraction, production and disposal, are modelled in the top layer. Each of these stages can be specified more exactly in their own sub-layer. Thus very long and complex processes can also be modelled and displayed in a clear way.

In assessing the life cycles of products the main focus is often placed on the output. The main question is: How can a certain amount of output (product) be produced with a minimum environmental impact (output-orientated calculation)? However, to assess other proceedings, other approaches are more appropriate. For example, in waste management the question “How can a certain amount of waste be treated with a minimum of environmental impact?” is of importance which is an input orientated approach. Good software tools offer the possibility of orienting the calculation towards any process within the process chain.

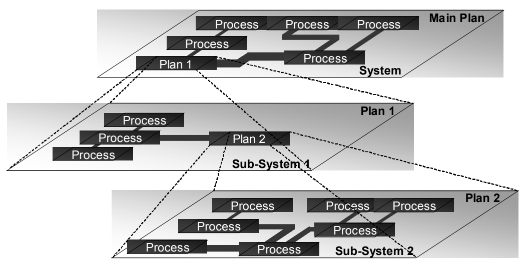

Figure 1. Schematic figure of a hierarchical structure [IKP, PE, 2002].

Some output-oriented software tools allow only one output of a process for the follow-up. Other outputs (by-products) are then addressed as negative inputs, which cannot be followed in the same process chain.

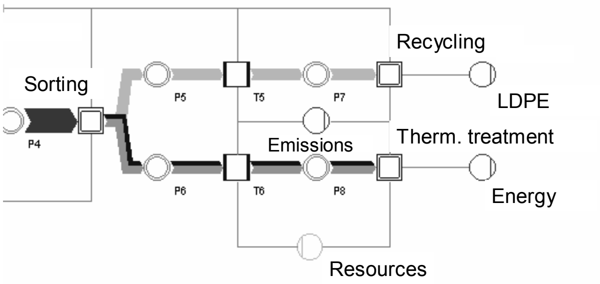

Figure 2. An example of a process chain. (Umberto)

But the user will sometimes encounter process chains with more than one output. Good software offers the possibility of following up different outputs. A simplified example from waste management is given in Figure 2. A sorting process has two outputs: LDPE films, which should be recycled (upper stream), and waste and other plastics that are designated for incineration (lower stream; the two colours indicate two different materials). The software enables the user to continue the process chain using both outputs.

4.2 Transparency, flexibility and user-friendliness

The structure of a software tool is partly responsible for its transparency. The calculation modus is important for a transparent compilation of an LCA. The user should be able to trace back each result in order to find mistakes. The proceedings from the modelling should be chronological and logical. Of vital importance in this context is the user interface, which should be clearly structured and self-explanatory. Modelling the process chain on a graphical interface is a very transparent way of modelling. There processes are arranged (e.g. as plan, network, assembly) and connected with material flows (arrows). The drag-and-drop feature is very helpful in this context.

To improve the user-friendliness many user interfaces have been designed similar to MS Office applications. The user will feel familiar with a number of features from the start. This improves the working quality of the software.

The user will often have to present his results to different groups of people (such as purchasers or the scientific community). Thus the software tools should offer a good presentation toolbox. A sankey diagram (see Figure 2) is a good option for presenting a process chain. The hierarchical structure can help to present the results clearly. Also it should be possible to create diagrams. A uniform layout for printed reports can enhance the software quality. Further the compatibility with other applications such as MS Office is important.

The implementation of a documentation feature is recommended to comply with the ISO 14040 standard, which defines the LCA.

The user should feel comfortable using the software tool. Little features, such as the possibility to change the entry unit, a zoom function for modelling the process chain or the possibility to show the input/output inventory in different gradations of detail, can make modelling more convenient for the user.

It is not easy for software developers to comply with often contradicting features such as user-friendliness and flexibility. Eriksson et al. [2002], for example, state that their software should be seen as a service rather than a computer program. Further they point out that they are continuously working to enhance user friendliness without losing flexibility. This is true for many software tools.

An important aspect in the context of user-friendliness is the time needed to learn how to use the software. The amount of time that has to be invested should be appropriate relative to the level of detail of the LCA.

4.3 Database

Data should be stored separately from the modelling module. It should be created and managed in some kind of database or library. This storage base has to be structured clearly. Very convenient for most users is a database structure similar to the one in MS Windows Explorer, where data can be edited without working in a modelled scenario. Also the import from and export to other applications is easier.

Apart from processes and flows, a database also contains modelled process chains. It should also be possible to file sub-layers in a process chain. They can be reused to model other scenarios. Further it should be possible to file separate data for a specific project so that the user does not need to search the entire database when looking for a specific process.

The data in the database need to be of good quality. They should be up-to-date and from a reliable source. More than one source is desirable in order to limit the danger of making mistakes. The user needs to clearly define the conditions under which the data are valid as well as the region for which they can be applied (e.g. energy for different countries). It can be helpful to include a data quality index to indicate the level of data quality.

An automatic update should be provided as soon as new data or data of better quality are available.

Good quality data should contain following information in the documentation:

- original data source
- age of the data
- composition of the data (number of companies or different literature, where the data are generated).

The user, particularly the scientist, will often use data from the database as well as from his own generated data sets. Processes and especially material flows have to be named carefully. Problems occur if different entries are created for one flow. For example, a process from the database produces the output “CO2.” Then the user creates a new process with the output “carbon dioxide.” The result is that two different names stand for the same flow. A feature that defines these flows as equivalent is necessary. This is especially important when the user creates his own data and the valuation. A very user-friendly way of communicating that two names stand for the same flow is to define synonyms.

Sometimes the user may want to connect processes where the output and the successive input are different. This should also be possible. An example of this is when the output of a process is “miscellaneous plastics” and the input in the next process is “waste” (plastics).

4.4 Calculation methods, uncertainty and variability analyses

Software tools offer different options for defining the proportion of inputs and outputs of a process. The simplest is to define a mass balance, e.g. the inputs are 1 kg of A and 2 kg of B and the output is 3 kg of C. However, mass balances are usually insufficient. Linear equation systems are an adequate way of modelling processes most of the time. Some tools also offer scripts, enabling the user to calculate non-linear systems like iterations.

Up to now LCA software tools have not usually considered the factors of uncertainty and variability. This refers mainly to parameter uncertainty (e.g. inaccuracy of emission measurements or of normalisation data) as well as the variability between sources (e.g. different emissions of comparable processes) and objects [Huijbregts, 2001].

The spectrum of tools to deal with these potential distortions ranges from simple parameter variations and sensitivity analyses to sophisticated methods, such as fuzzy logic computations, Bayesian statistics or probabilistic simulations.

In particular, simulations based on statistical modelling methods seem to be a promising technique for making uncertainty operational. Two approaches – the Monte Carlo and Latin Hypercube simulations – are currently implemented in LCA software tools [Weidema and Mortensen, 1997].

To perform the Monte Carlo simulation, the uncertainty distribution (normal or rectangular are usually available) of each parameter has to be specified. All the parameters vary randomly within the limits of the given distribution. The randomly selected values are inserted in the output equation. After repeated calculations, the output is represented by a predicted distribution of each output parameter. The Latin Hypercube simulation works in similar way. The main difference is, that the uncertainty distribution of a parameter is segmented in a number of non-overlapping intervals with equal probability. This fact leads to generally more precise random samples than the Monte Carlo simulation [Huijbregts, 2001].

In LCA practice the application of these methods is useful for assessing the influence of the parameter uncertainty on the uncertainty of the model output. The most important consequence of such analyses is the identification of parameters that cause a large spread in the model output. This can help to increase the accuracy of the overall model.

4.5 Methodological Properties

For waste management questions LCA normally leads to the comparison of different treatment options for waste streams with a reference scenario (e.g. landfilling) that provides a functional equivalence. This equivalence can be achieved either by given credits outside of the system or by expanding every system to achieve the same benefits. To use the LCA software tool comfortably it is necessary to provide both methods (the “credit method” and the “basket-of-benefits” method). Especially complex scenarios cannot really be addressed with the basket-of-benefits method. If only “credits” can be provided by inverting existing primary production processes, the assessment will not be comfortable, because outputs are shown in the input table and the other way round. In a good software credits are automatically subtracted from the outputs.

At the international level two impact assessment methods have been established and are most commonly used in Life Cycle Assessment: an operational guide to the ISO Standards (CML 2001 method [Guinée et al., 2001] and Eco-

Indicator 99 [Goedkoop et al, 2000]). Less often used methods, particularly in the German language area, include the Swiss Eco-factors 1997 [BUWAL, 1998] and the German Federal Environmental Agency (UBA) method [UBA, 1999]. The software should at least provide both internationally used methods because they follow different general approaches: problem-oriented methods (CML) and damage-oriented methods (Eco-indicator).

Especially when the CML method is used for the impact assessment, the software needs to provide another aid for interpretation of the results. Weighting the results according to their relative importance often is necessary for the results interpreter. One possibility for results aggregation is normalisation, where calculating the magnitude of indicator results relative to reference information is possible. The software should provide different normalisation parameters.

In general a different quality of results should be given, e.g. a thorough inventory, different valuation results, aggregated values of different impact categories, or a summarisation to just one parameter to afford a ranking of options.

4.6 Service and Support

Service and support are very important aspects of LCA software and should not be underestimated. Software needs continuous maintenance.

The database especially needs a great deal of attention to keep it up to date. The software should also be continuously improved to eliminate malfunctions and improve user-friendliness and software ergonomics.

A telephone or e-mail hotline should be provided to ensure that the user receives qualified help for technical as well as methodological problems. A detailed manual is essential. Many LCA software providers offer special training sessions to introduce the software to the new user. Demonstration versions and tutorials to demonstrate the functionality and features are very helpful in providing a quick overview of the properties of a software tool. Such demo versions should be available for free to demonstrate the advantages of a software tool to potential new users.

Another essential aspect of service is getting relevant information about the software. This aspect should take into account that there are at least two different kinds of users. On one hand there is the LCA newcomer: He needs some general information about LCA and about the advantages of the particular software. This information can normally be found on the software homepage. On the other hand there is the professional LCA user: He needs more detailed information about the different features the software provides and the assumptions included in the database or the methodological solutions, such as which assessment methods are provided and where the database is from. At resent there is a lack of information in this area. Normally one sees this information only after purchasing the software. More detailed information is needed on the Internet for LCA software.

4.7 Other features

As mentioned before, many LCA software tools offer additional features. One group of them focuses on analysing data. One example is a sensibility analysis, which should be implemented in each good software product. The feature of comparing different scenarios can also be called a standard feature.

The cost consideration is also important. Although there are major methodological differences between Life Cycle Cost analysis (LCC) and an LCA, they can be tightly, logically and practically integrated with one another [Norris, 2001]. Some software tools also consider time aspects and social parameters such as working time.

5 CONCLUSION

Many LCA software tools can be considered of good quality. They were often developed for a specific application of LCA but were then improved for a wider scope. Sometimes although the software is generally designed for a wide scope, it is not possible to use this wide scope due to e.g. inadequate calculation methods or an unsuitable structure. Thus it is not enough for single features to be implemented in an LCA tool, but the whole package of features needs to fit together in a good quality software tool. Basic requirements need to be fulfilled by the software to be suitable for a wide audience.

Generally a software tool should operate smoothly and quickly, without errors due to mistakes in the software programming. The hardware requirements should also be adequate. A hierarchical structure is essential for good quality software, in order to be able to work on more complex problems as well. A clear structure ensures transparency and modelling comfort. The starting point of the calculation should be of free choice. Also the modelling of different outputs should be possible. The results should be transparent. A graphical modelling of the process chain is very convenient for the user.

Compatibility of the software with other application should be provided and the user interface should be designed in such a way that the user finds his way around easily and feels comfortable working with the program (e.g. if designed similar to MS Office applications). A good toolbox to present the results is desirable.

The database should be managed and edited separately (creating, deleting, modifying of data). The data should be of good and transparent quality. There should be a possibility of separately saving and organising data used for single projects. The names of processes and materials need to be clear and logical and the problem of synonyms should be taken into consideration.

It should be possible to choose between different methodological approaches for the impact assessment and the aggregation of results as well as for the comparison of scenarios with different outputs.

An Internet homepage with detailed information should be provided for an LCA software tool. It should contain information for newcomers as well as experts. Different versions and a free demo version of the software should be available.

Additional features that help the user to analyse results and allow further calculations are important requirements for some users.

To define the proportion of input and output, linear equation systems will most often be sufficient, although scripts can be essential for some processes.

Good software tools featuring uncertainty and variability analyses such as the Monte Carlo simulation enable the user to identify parameters, which cause a large spread in the model outcome. Thus the accuracy of the model can be increased through support of a more selective procedure.

It is important that an LCA software tool be continually improved and updated with new developments in the field of LCA. Maybe they can even give an incentive to new developments since most life cycle assessments are calculated with an LCA software tool.

6 REFERENCES

BUWAL, Bewertung in Ökobilanzen mit der Methode der ökologischen Knappheit – Ökofaktoren 1997. Schriftenreihe Umwelt Nr. 297, 1998.

Eriksson O., B. Frostell, A. Björklund, G. Assefa, J.-O. Sundqvist, J. Granath, A. Baky, and L. Thyselius, ORWARE – A simulation tool for waste management, *Resources, Conservation and Recycling,* 36, pp. 287-307, 2002.

Frühbrodt E., LCA Software Review – An up-to-date overview of the European market, available on http://europe.eu.int/comm/enterprise/electr_equipment/eee/workshop9-10-02/present/lcasoftwarere.pdf .

Goedkoop M., S. Effting and M. Collignon, The Eco-indicator 99 – A damage oriented method for Life Cycle Impact Assessment. PRé product ecology consultants, second edition, 2000.

Guineé J., M .Gorrèe, R. Heijungs, G. Huppes, R. Kleijn, A. de Koning, L. van Oers, A.W. Sleeswijk, S. Suh, H.A.U. de Haes, H. de Bruijn, R. van Duin, M.A.J. Huijbregts, E. Lindeijer, A.A.H. Roorda, B.L. van der Ven and B.P. Weidema, Life cycle assessment – An operational guide to the ISO standards. Ministery of Housing, Spatial Planning and the Environment and Centre of Environmental Science - Leiden University, 2001.

Huijbregts, M.A.J., Uncertainty and variability in environmental life-cycle assessment, thesis, University of Amsterdam, 2001.

IKP, PE, GaBi 4 Software-System and Database for Life Cycle Engineering (User Guide) Stuttgart, Echterdingen 1992-2002.

Jönbrink A.K., C. Wolf-Wats, M. Erixon, P. Olsson and E. Wallén, LCA Software Survey, IVL rapport No. B 1390, IVL Svenska Miljöinstitute AB, Stockholm, 2000.

Norris G.A., Integrating Life Cycle Cost Analysis and LCA, *International Journal of Life Cycle Assessment*, 6(2) pp. 118-121, 2001.

Rizzoli A.E. and W.J. Young, Delivering environmental decision support systems: software tools and technique, *Environmental Modelling & Software*, Vol. 12(2-3), pp. 237-249, 1997.

UBA (Umweltbundesamt), Bewertung in Ökobilanzen, Version 1999, UBA Texte 92/99, 1999.

Unger N, Eignung unterschiedlicher Softwaretools für Ökobilanzen bei abfallwirtschaftlichen Fragestellungen, diploma thesis, Boku-University, Vienna, 2003.

Weidema B.P., and B. Mortensen, Results of a test of LCA-software with statistical functionality, Note for the Danish Environmental Protection Agency, Institute for Product Development, Lyngby, 1997.