Int.J. Applied Thermodynamics, Vol.4, (No.2), pp.77-83, June-2001

ISSN 1301-9724

Methodical Design Support for Life Cycle Assessment-Design*

Wim ZEILER
Kropman B.V.
Mechanical & Electrical Contracting
Verrijn Stuartlaan 36
2288 EL RIJSWIJK - The Netherlands
E-mail: w.zeiler@ry.kropman.nl

Abstract

Decision support is needed within design processes in order to take into account all the aspects of life cycle assessments (LCA). In order to develop appropriate tools for decision support in design processes, it is necessary to found them on an understanding of design. The framework of methodical design as developed at the University of Twente is extended to integrate the specific LCA aspects. The design process is being described in terms of several major phases, levels of abstraction and connected levels of modeling centered on layered design representations.

Key words: Life cycle assessments, methodical design

1. Design: An Introduction

Design, optimizing and integrating LCA aspects are still difficult to achieve with much ease. The environmental impact of the product, which still has to be designed, has to be taken into consideration without knowing quantified specifications.

With environmental aspects involved from the beginning of the design process, the number of uncertainties becomes overwhelming (Toxopeus and De Jong, 1999). In this paper, LCA is structured in view of its use in decision making. Emphasis is put on often-encountered inconsistencies, namely the set up of LCA-system models, the representation of decisions and value choice of actors involved in a product system and the representation of changes within the economic and cultural system.

As the design process proceeds, so the information about the product being designed increases. The design process may, therefore, be viewed as progressing from an information poor condition to one that is information rich. The richness of information relates not only to the quantity of information but also to the quality of understanding of the relationship between the elements of information (McGown and Green, 1999).

Decisions made during design are known to influence the performance of other life-cycle phases.

Decisions that seem good for one phase can lead to problems and high costs in another. Ideally, designers should have an insight into how their decisions influence the different life-cycle phases. The main motivation behind the ideas presented in this paper is to provide designers with knowledge that allows them to foresee the consequences in multiple life-cycle phases resulting from their decisions. Research carried out so far indicates that there is a lack of formal understanding of the origin, type and content of the relevant knowledge relating product design characteristics with life-cycle consequences and how these interact. One of the critical problems in engineering design is making early decisions to satisfy a host of total life-cycle phase issues. High costs and delays due to problems arising in manufacturing and other life-cycle phases from design phase decisions is the specific motivation behind this approach. This is evident, still designers have difficulty in taking into consideration the life-cycle consequences (and their interactions) of their design decisions (Borg, 1999).

---

* This paper was presented at the ECOS2000 Conference in Enschede, July 5-7, 2000

There are a number of manual and computer-based design tools that are employed to avoid the generation of life-cycle problems, such as FMEA (Failure Modes & Effects Analysis), DfA (Design for Assembly) and DfM (Design for Manufacturing) (Veefkind, 1999).

However, they offer only a narrow view to the consequences and they are mostly employed toward the later design stages when major design decisions have already been committed. Most robust ideas come from applying, however, Design for Principles (Goulding, 1997); cannot be used for the most important early stages of the design process. On the other hand Concurrent Engineering, or more correctly concurrent product development, requires designers to take a multiple life-cycle phase view toward their evolving design solution. An emerging problem to such an approach is the generation of models of interrelated life-cycle consequences as a means to life-cycle providence during the early design stages.

It is believed that life-cycle providence can be employed for supporting the exploration of the design phase without necessarily constraining it.

This objective can be achieved through (Borg, 1999):

- The identification of design decision factors that influence the consequences generated during the different life-cycle phases;
- The structuring of the knowledge which links design decisions to their consequences or behavior;
- The modeling of the mechanism through design decision consequences propagates through different life-cycle phases.

The environmental issue is one of the most important and a critical issue to be tackled urgently. The modern industrial system has produced an enormous amount of products to try to satisfy both the need from the marketplace and the profit of enterprise. The need for the management of a product life cycle, however, has not been recognized as much as the need for production. One of the technical reasons for the weak recognition in the manufacturing section about product life-cycle management is that there has not been an effective way for designers to communicate with specialists or users. In order to manage the product life cycle, designers have to collect various inputs from specialists because the environmental issue covers many disciplines. It is also important to let users know the design concept including life-cycle management so that they can use the product satisfactorily. It is a challenge for design engineering to provide a method to plan the life cycle of products better and easier (Kurakawa, et. al. 1996).

The discipline of design, unfortunately, seems to contain rather a lot of the world's complexity. Design is multidisciplinary, nested, sometimes intangible, repetitive and iterative. It can involve poor logic, uncertainty and paradox (often at the same time). It is important to have the right tools for the designers so they can do their job with much more ease and confidence. A way to solve this complex interaction between design, environment and individual users is to look closely at the key factors of design. The general aim of design is "structuring the known data of nature in a way leading to an effective control of matter in relation to man's needs and wants" (Kroonenberg, 1986). A methodology will bring structure and organization to the problem, helping to keep the complexity to a manageable level. It can also help by guiding towards a constructive way of thinking about a problem.

2. Conceptual Model

Through the different levels of abstraction, the product model is gradually described in an increasingly more detailed manner. The different levels of abstraction should be considered as a representation of a particular view on the total information available for a design.

This integrated product model must:

- be able to differentiate related information.
- support differentiations related to the different levels of abstractions (views) by being structured into corresponding sub-models.
- ensure the satisfaction of consistency and completeness constraints linking different levels of abstraction in the design process.

Design, as a solution-evolving process, involves activities of searching information, analyzing, manipulating and structuring information about the problem to be solved. Generating new information and evaluating and communicating information is a major activity within the process. In the future, information that is useful in designing will be available in quantity and quality heretofore not possible.

One of the major decisions to be taken is the choice of a design paradigm. A paradigm refers to a principle, or a set of principles, which is based on a theory or a methodology. It will provide the general knowledge needed for the design process. As a design paradigm, the Methodical Design Method is chosen (Kroonenberg, 1978). The design approach, developed and taught at the University of Twente in the Netherlands since the late sixties, was compiled and formalized in the seventies and elaborated (Boer, 1989). It is based on systems theory.

The use of a methodical design process method is a necessary guideline for the designer and makes it possible to structure the design process. The methodical design approach has been selected for three reasons: (1) it is a problem-oriented approach; (2) it is the only model emphasizing the execution of the process on every level of complexity; (3) it is one of the few models explicitly distinguishing between stages and activities (Zeiler, 1996).

The methodical design process can be described on the conceptual level as a chain of activities which starts with an abstract problem and which results in a solution. Four main phases are distinguished in which eight levels of functional hierarchical abstraction, stages can be distinguished. A feature of the used extended model of methodical design is the occurrence of a four-step pattern of activities in each stage. In systems theory the same activities are proposed for decision processes as can be found for the design process. The four phases, with their specific level of abstraction, have their own requirements for the descriptive model of decision making, each defining its own associated inference to the aspects of LCA. Some suggestion on how to take the requirements into consideration will be given.

3. Functional Decomposition

In order to survey solutions, engineers classify solutions based on various features. This classification provides means for decomposing complex design tasks into manageable size problems. An important decomposition is based on building component functions. The functional decomposition is carried out hierarchically so that the structure is partitioned into sets of functional subsystems. The decompositions are carried out until we arrive at simple building components whose design is a relatively easy task.

Hierarchical abstraction means the decomposition of information into levels of increasing detail, where each level is used to define the entities in the level above. In this sense each level forms the abstract primitives of the level above. These terms in the upper level form condensed expressions of a given relational and/or operational combination of primitives from the level below.

The sets of generic primitives are located at distinct levels of abstraction, ranging from the system level to the physical level. The contents of the layers are based on the technical vocabularies in use, therefore we will speak of technology-based layers or levels. Each layer represents an abstraction of the levels below. For a more extensive description of the models that formed the basis for the notion of technology-based layers (Alberts et. al., 1992).

It is important to realize that the actual contents of the layers as well as the number of layers will be domain-specific.

Generic components represent behavior that is known to be physically realizable. They are generic in the sense that each component stands for a range of alternative realizations. This also implies that the generic components still have to be given their actual shape. Relevant technical or physical limitations manifest themselves in the values of a specific set of parameters belonging to the generic components. These parameters are used to get a rough impression of the consequences of certain design choices at the current level of abstraction for the final result. In most practical situations, purely hierarchical design is an illusion due to the interrelations between the elements of a decomposition.

Discarding the dependencies among subparts will often result in conflicting mutual requirements between the parts during recombination (Alberts,1993). The separation is made between:

System Theory	Activity	Methodical Design
Definition Demands	Generate	Problem definition
Synthesize Analyze	Synthesize	Working principle
Evaluation Decision	Select	Detail design
Implementation Application	Shape	Realization

- Information level, knowledge orientated, representing the "conceptual world".
- Process level, process orientated, representing the "symbolic world".
- Component level, device orientation, representing the specification world. This makes four levels of aspect abstraction in the descriptive model of design, where each defines its own ontology and associated inferences

4. Information Level

This level deals with the knowledge of the systems by experts. One of the essential ideas behind this is that human intelligence has the capability of search and the possibility to redirect search. This information processing is based on prior design knowledge. One of the major problems in modeling design knowledge is in finding an appropriate set of concepts that the knowledge should refer to, or, in more fashionable terms. an ontology (Alberts, 1993).

5. Process Level

This level deals with physical variables, parameters and processes. The set of processes collectively determines the functionality of the variables that represent the device properties Modeling at the functional level involves deriving an abstract description of a product purely in terms of its functionality.

This abstraction reduces the complexity of engineering design down to the specification of the products desired functionality.

6. Component Level

This level describes the hierarchical decomposition of the model in terms of functional components and is domain dependent. Generic components represent behaviors that are known to be physically realizable. They are generic in the sense that each component stands for a range of alternative realizations. This also implies that the generic components still have to be given their actual shape.

7. Part Level

This level describes the actual shape and specific parameters of the parts of which the components exit. Relevant technical or physical limitations manifest themselves in the values of a specific set of parameters belonging to the generic components. These parameters are used to get a rough impression of the consequences of certain design choices at the current level of abstraction for the final result.

The design problem as defined here is how to obtain a physical description of a technical system given a high-level, but abstract, specification. Theoretically, one can envision a continuous decrease in abstraction ranging from one extreme to the other. In practice, however, design will never be of a completely top-down form. In order to determine what the next refinement action will be, a designer has to have knowledge of what the possibilities roughly are. Knowledge about the possible functions realized, given specific physical properties of the realization material, is propagated upwards. In the framework presented here, this bottom-up knowledge is represented as sets of generic components (that are known to have several possible physical realizations and representations) at different levels of abstractions. The design process can be described in terms of transformations of the design description within or between the technology-based layers. The behavioral description of the required system at a certain level is decomposed according to the behavior of the available generic components. As a result different configurations may arise. In order to estimate what the restrictions imposed by generic components imply for the overall system, the resulting configuration has to be parameterized in terms of the form characteristics of the individual components. Throughout this process, the performance requirements serve as constraints on the possibilities for configuring generic components into larger structures and for assessing the physical characteristics

8. Engineering Model Libraries

The Methodical Design approach depends on the availability of a collection of sub-models, and it provides the organizational framework required for structuring and maintaining such a collection. Model libraries can be used to share and reuse knowledge that is not part of an abstract theory (such as physics) but still generic across different applications. A model library is structured along two dimensions. First, there is a part of the hierarchy that is defined by the decomposition of functional components. Every functional component knows about one or more decompositions to its immediate sub-components and about the super-components it can be a part of.Second, models are ordered in a taxonomy based on invariant properties. More detailed models (that is, models based on more detailed assumptions) can inherit general properties from less detailed ones.

Design level	Main topic of abstraction	Output	Focus from LCA point on
Problem definition	Information "conceptual world"	Need Design problem	Environment
Working principle	Process "symbolic world"	Functional specification Physical solution process	Energy Exergy
Detail design	Component "real world"	Module structure Prototype structure	Re-useability
Realization	Part "specification world"	Material properties	Resources

This framework is used in practice as a basis for organizing the library to be developed. A complete model - i.e. a model for a particular device within a certain task environment - is composed by selecting a library component from each level, while keeping the references to possible alternatives.

9. EBIB

Creating mathematical models is time consuming when one has to start from scratch for every new application. However, many similar components will occur frequently so their models can be expected to be corresponding.

In the EBIB. Dynamic Model Components for a Heating Systems Library Project, Kropman B. V., The University of Twente and the Netherlands Energy Research Foundation, ECN, worked together to study the possibilities for building a library of model components. A library of model components is developed in the EBIB project in order to support model-based design and diagnosis of systems for central heating and hot water supply. With these models the expected behavior of designed installations can be studied. In that way, it is possible to check the proposed system behavior against the given requirements and to generate alternatives. A simple central heating system is described in terms of a functional components model and a bond-graph model. The EBIB-project provides an initial set of library model components that is expected to be adequate for the typical HVAC configurations handled by Kropman B.V.

10. OLMECO

The University of Twente and the Netherlands Energy Research Foundation ECN developed re-usable and sharable model components in the thermodynamic domain for the OLMECO design library. The OLMECO project is an Esprit III project for building an Open Library for Models of mechatronic Components. The core of the OLMECO software is a conventional (OO/relational) database for storage and retrieval of mechatronic model fragments; we will give an impression of its structure by considering the most important parts of the conceptual scheme of the database (Top et. al., 1994), for some examples of its use (Top et. al.,1994).

Systems can be decomposed in different ways; functions of which components are the carrier can be realized by different physical processes, and physical processes can be specified by different mathematical constraint expressions. The proposed generic structure of the OLMECO library has two important advantages:

1. It separates different groups of modeling decisions, thus giving support to the user support and facilitating a piecemeal approach to engineering model construction, and
2. It provides a breakdown of stored models into parts that have a generic nature, thus enhancing reusability and shareability of library models.

A component taxonomy can be stored in the library by means of the kind of generalization/specialization relationship. This information can be used by the modeler to quickly access the components he or she wants to use.

11. A Modeling and Simulation Experiment: The Schieland Hospital Heating System

To test the usefulness of the OLMECO library, UT and ECN have contributed to the library with thermodynamical models for components like pipes, valves, splitters and mixers, heaters and heat exchangers. The modeling experiment for the thermodynamic domain consisted of the modeling and simulation of a large central heating system. This section describes this experiment. During the experiment there were two questions in mind:

a. The practical usability. The subject of the experiment is the modeling and simulation of the existing heating system of the Schieland Hospital, a general hospital in Schiedam, the Netherlands.
b. The model contains a large number of components from the thermodynamics library and is mathematically complex because of the structure of the system and the fact that both hydraulic and thermodynamic behavior is modeled. For a detailed description of this, see (Top et. Al., 1995).

12. Simulation Results

The first conclusion that can be drawn from this experiment is that the OLMECO library provides good assistance in the modeling process. This is reflected in the amount of time it took to construct a large and complex model like the one described in this section and the quality of the results.

Modeling the system took a short time due to the fact that the library contained most of the required model fragments and to the fact that the model could be specified incrementally, starting with the component model that is very similar to the schematic drawing of the system. The other steps to processes and mathematics were guided very well by the suggestions the library contained for possible process descriptions and mathematical relations. The second conclusion is that the thermodynamic library is diverse enough to support compositional modeling of real world systems. The modeled system is considered to be large and contains a variety of components typical for the whole domain.

13. Conclusion

Methodical design is proposed as a theoretical basis for decision support in design processes. Design is viewed as a problem-solving activity in which functional reasoning is central.In order to allow a stepwise approach in which each design decision has well-defined implications on the integration of the specific LCA aspects, four different ontological levels have been distinguished within design process:

* Specified Parts
* Functional Components
* Physical Processes
* Information Model

These levels provide a structured framework for model libraries, which in addition could contain separate entries for observation data for validation purposes about LCA aspects. Methodical design can be used as a formal modeling technique. The benefits of methodical process design strongly depend on the availability of generic models stored in structured libraries. Model libraries based on the approach presented here will significantly advance the state of automated design process.

Acknowledgements

This paper is based on the work of Top, Borst, Akkermans (1995) and their previously published papers, and has been supported by Senter. The partners in the EBIB project are Kropman B. V., ECN and the University of Twente. The EBIB project has benefited from earlier work in the project. The partners in the OLMECO project are PSA Peugeot-Citroën (F), BIM (B), Fagor (Sp), Ikerlan (Sp), Imagine (F), University of Twente (NL) and ECN (NL).

References

Alberts L. K., Wognum P. M., Mars N .J. I., 1992, "*Structuring Design Knowledge on the basis of Generic Components, Artificial Intelligence in Design*", J. S. Gero (ed.) Kluwer Academic Publ..

Alberts L. K., 1993, YMIR: "An Ontology for Engineering Design", PhD-thesis, University of Twente, Enschede.

Boer, S .J. de, 1989, "Decision Methods and Techniques in Methodical Engineering Design", PhD-thesis, University Twente, ISBN 90-72015-3210, Faculty of Mechanical Engineering.

Borg, J. C., 1999, "Knowledge Intensive Component Life Design Synthesis", University of Malta www.cad.strath.ac.uk/~jonathan/Home_res.html

Goulding, J. R., 1997, Design Methods: Design for X, Engineering Management Program, Port State University.

www.emp.pdx.edu/Searchable/Std_projects%20 PDF%files/emp-9771/De.../DFX.ht

Kinetic, L.L.C., 2000, FMEA Methodology. http://www.fmeca.com/ftfmethod/methodol.htm

Kroonenberg, H. H. van den, 1986, "CAD applications in the creative phases of the methodical design process", *Proceedings CAPE '86*, Copenhagen, May 86, Elsevier North-Holland, Amsterdam.

Kroonenberg, H. H. van den, 1978, Methodisch Ontwerpen (WB78/OC-5883), University of Twente, Faculty of Mechanical Engineering.

Kurakawa, K et. al., 1996, The Green Browser: An information staring tool for product Life Cycle Design, The University of Tokyo, 27 august 1996
www.race.utokyo.ac.jp/~kurakawa/papers/dtm96.html

Mc Gown, A D., Green G., 1999, Recording The Design Process in Real Time: Innovate Design Concept Synthesis, Dept. of Mechanical Engineering, Glasgow University, Scotland, UK www.dmu.ac.uk/ln/4dd/synd1e.html

MVAC, 2001, Michigan Virtual Automotive College. Design for Assembly, http//engineer.gvsu.edu/vas/

Toxopeus, M. E., de Jong, J. J., 1999, Life Cycle Oriented Designing, University of Twente, Faculty of Mechanical Engineering M.E.Toxopeus@wb.utwente.nl

Top, J .L., Breunese, A. P. J., Dijk, J. van, Broenink, J., Akkermans, H., 1994, Conceptual schema of the OLMECO library. OLMECO deliverable, ESPRIT project 6521 OLMECO/WP3.3/ECN/01/2.0, ECN and University of Twente.

Top, J L., Breunese, A. P. J., Broenink, J .F., and Akkermans, J. M., 1995, “Structure and use of a library for physical systems models”, *In Proceedings International Conference on Bond Graph Modeling and Simulation IC BGM'95*, Las Vegas 15-18 January, SCS.

Top, J. L., Borst, P. and Akkermans, H., 1995, Reusable thermodynamic model components for design. OLMECO deliverable, ESPRIT-III project 6521 OLMECO/WP2T45/ECN/01/4.0, ECN and University of Twente, November 1995.

Veefkind, M. J., 1999, Design for Manufacturing.
http//www.io.tudelft.nl/education/ide342/IDE342_3/index.htm

Zeiler, W., 1996, “An intelligent computer aided process control approach to design building installations”, *The 3rd International Conference on Concurrent Engineering Electronic Design Automation*, 10-12th April, Cambridge, UK.