Modelling and Simulation in LCA

Peter Forsberg

Environmental Systems Analysis (ESA)
Centre for Environmental Assessment of Product and Material System (CPM)
Chalmers University of Technology
Göteborg
Sweden

Acknowledgment

I would first like to thank Ann-Marie Tillman for being the project leader for this study and taking the time to participate in almost endless discussions on the need for calculations in LCA. Raul Carlson for helping me in sorting out basic concepts in the academic world in general and LCA in particular. Karin Gäbel for providing me with an intriguing and tricky cement process to model. As it turned out I learnt a lot doing so. Sverker Molander for trying to help bring order and direction to a, by that time, very vaguely formulated project. Torsten Wik for taking his time to read trough this report and discussing the mathematical aspects of the contents. Margareta Wester for believing in the basic project idea and supporting the continuation of the project.

Thank you!

Summary

LCA (Life Cycle Assessment) is an environmental evaluation technique that has expanded both in importance and complexity since it was introduced in the early 90's. At the core of it is the model over the technical system and the calculations made on it, i.e. the Life Cycle Inventory Analysis (LCI). It is thus related to the area of modelling and simulation. This report introduces basic concepts and recent research results derived from that area, presented from an LCI point of view. It explains the models and calculations performed in LCI using the above concepts and nomenclature and gives some proposals on how to enhance and simplify the process of creating models and performing simulations in LCI.

Glossary

Algebraic loop: A function requiring its output as input
Closed loop: Any type of feedback to earlier parts in the causality chain
Experiment: An experiment is a test carried out on a system or model of a system
Inconsistent loop: Feedback that results in a mathematically over-specified problem
Model: Simplified (here mathematical) description of reality used for simulations
Optimisation: Finding the optimal point (max or min) evaluated by the target function by varying one or more variables
Over-specified problem: A problem with more constraints (equations) than variables
Simulation: Experiment (calculation) performed on a model
System: A clearly defined and limited part of reality to be studied
Under-specified problem: A problem with more variables than constraints (equations)
Well-posed problem: A problem (equation system) resulting in a square matrix

Contents

1 Introduction

This is a report for the feasibility study to “Modelling and Simulation in LCA”. The objectives of the feasibility study are to find how LCA relates to general modelling and simulation and to investigate other engineering areas for usable concepts, techniques and possibly tools.

Nowadays no matter which engineering area one wants to explore, there is an abundance of knowledge available. Modelling and simulation is a very wide engineering area of which the importance has increased over the last few years. The reason is the rapid development of powerful computers to perform more advanced types of simulations. Now it is possible to digitally simulate systems that were almost impossible to deal with before. Besides, as the development goes on even more complex systems can be calculated. As a result more research effort has been put on finding ways to perform the modelling to render a more flexible model concerning both modularity and types of applicable simulations. These recent findings have been well adopted in most of the related applied engineering areas, such as control theory, electrical circuit simulation etc.

Modelling and simulation in LCI has not, however, made use of these recent findings yet, in spite of there being a great potential in doing so. That involves not only increasing the number of available evaluation techniques, but also to learn and adapt nomenclature from related areas. This will expand the usability, understandability and general acceptance of LCA as well as LCA in a wider perspective. Another aspect is that already available software systems may be used within the specific area of interest, which may result in saving of effort.

The chapter “Important concepts in simulation?” gives a short definition of the most important terms used in modelling and simulation. Nomenclature and methods that can be used within LCI are explained and examined. The mathematical definitions and in-depth explanations are presented in appendix 1, “Simulation properties”. The next chapter, “Simulation in LCI”, shows what this means in terms of LCI, the present most common approach, and, in addition, how it could be extended to a more general one as part of future studies. In appendix 2, “Survey and categorisation” some examples of usable calculation techniques and software tools are gathered and categorised.

2 Important concepts in simulation

Techniques for modelling and simulation are older than engineering. The very process of formulating mathematical expressions and equations that describe reality into any form is an act of modelling. For example, all mathematically formulated physical principles are models, since they are simplified descriptions of the behaviour "real" physical world, and have the mere purpose of predicting the behaviour of that same "real" physical world, i.e. by mapping a certain behaviour realm onto mathematical form. Nevertheless, this is not what is here meant by a model. Instead a model is referred to as a well-defined description of a specific system that is derived from applying physical laws and formulas (empirically well-tested models) or some other understanding of the underlying phenomena. This is the meaning of the term modelling that is used throughout this text.

Modelling is the art of building models that represent reality. Experiments are then performed on these models to gain information about the behaviour of the underlying real system, as indicated in figure 2.1. A model can be anything that explains how something behaves; it can be a mental model over the behaviour of a certain person or a verbal model over how the world economy reacts. The type of models studied in this report are all mathematical models, i.e. models that can be described in mathematical form and coded into computer programs.

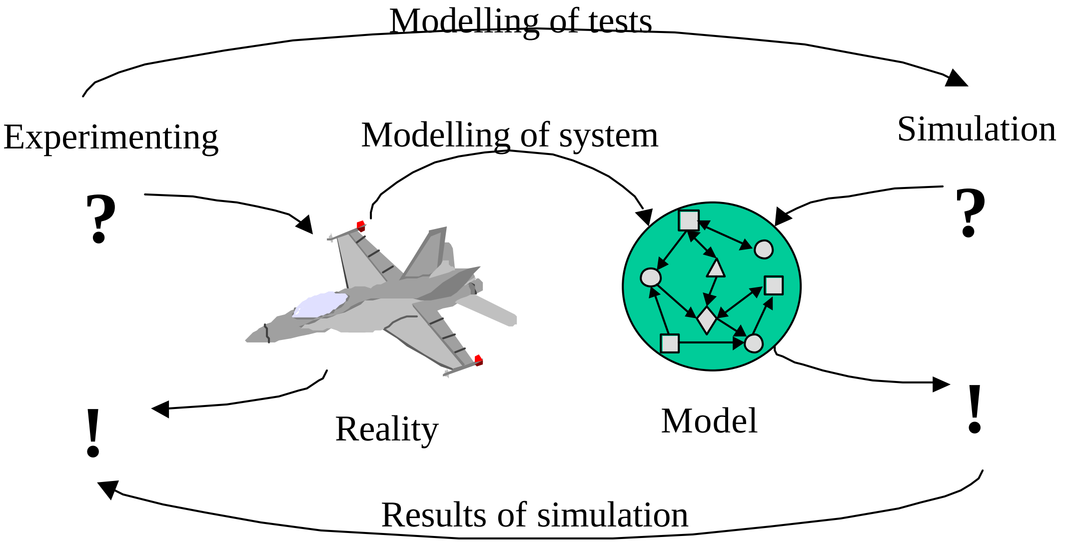

Figure 2.1. Modelling and simulation of real systems.

Four words are used in this text to describe the process of attaining information from reality and performing calculations on it. They are respectively *System*, *Experiment*, *Model* and *Simulation*. Even if they are commonly used throughout the literature, they often cause misunderstandings since virtually every textbook uses them in a slightly different connotation. The following explanation is denoted to clarify how they are used in this text. The nomenclature mostly complies with the one commonly used in system modelling and control theory. It is not, however, intended to give an in-depth treatment of the subject. Those who need to gain a wider perspective are encouraged to read some introductory books on the subject of modelling and simulation, e.g. [Cellier, 1991], [Bossel, 1994] or [Rosenberg, 1983].

2.1 System

Before a model can be built we need to specify a number of things about the piece of reality we wish to study. To start with the question of how far the system shall be extended, i.e. where to put the boundaries, is relevant. It is a matter of the scope and the purpose of our interest in the system. Also what interactions with the surroundings over the system boundaries we are interested in needs to be specified. The result is a well-defined piece of the real world with a well-known interface. This piece is called a *system*.

A system can thus be defined as:

> “A clearly defined and limited part of reality with defined interfaces that is to be studied”

With this definition a system can in turn be subdivided into any numbers of sub-systems, which all are true systems in themselves. On the other hand, it can also be extended to include the whole universe. Regardless of its size, it is necessary to formulate what aspects of the system and from what point of view it is to be evaluated. Even the simplest system, if such really exists, can be viewed from a practically unlimited number of points of view. The purpose of the study, i.e. what we expect to get out of if, defines the main focus for the system.

2.2 Experiment

Our interest in the piece of reality we defined as a system is that we want to exert it to input of some sort and observe the reactions, or outputs generated. This test can be performed on either a system or a model of the system. A definition might then be:

> “An experiment is a test carried out on a system or model of a system”

Sometimes experiments are carried out on real systems, i.e. in reality, for example when the behaviour of the system is not known or is too complex to imitate. It is usually, however, problematic to ensure there are no external influences that might disturb the test and induce unexpected and even false results. Another disadvantage is that some experiments may not be available since they are too costly, too dangerous or the time constants are too long to allow for the outputs to be observed. One of the most important advantages, on the contrary, is the validity of the system. The system is valid for all types of experiments, regardless the initial intention when the system was defined. The most usual types of experiments are though performed on models of system. Experiments on models are called simulations.

2.3 Model

A real system is part of the ‘real’ physical world, which may be transformed into a simplified and, in our context usable, form called a model. This definition was formulated by Marvin Minsky [Minsky,1965]:

> “A model (M) for a system (S) and experiment (E) is anything to which E can be applied to answer questions about S”

This is a very broad formulation where the ability to denote something about the underlying system is the important aspect. It can thus be anything from a general understanding of a situation to, which was common before the computers emerged, a for the specific simulation purpose specially made electrical circuit. In this context we will, however, continue with mathematical models, that is models that can be expressed in mathematical form.

Since the model is a simplification of reality, it is not valid for all types of experiments. An “experimental frame” can then be employed to link models and experiments in the proper way.

2.4 Simulation

A simulation is similar to an experiment, except it is performed only on models. It means to perform any kind of calculation with the data and structure provided by the model. This definition was coined by Korn and Wait [Korn, 1978]:

> “A simulation is an experiment performed on a model”

3 Simulation in LCI

In this chapter the current status and development on simulation in LCI is presented. It gives an indication about what kind of models are used, described from the system analytical perspective developed so far in this report. Then important simulation methods are mentioned and for the most important one, system normalisation to the reference flow, an example is given. It will also deal with some special problems that may occur when performing simulation in LCI.

Not much have been done in defining overlying simulation concepts in LCI. The overlying concept states how we chose to view the particular part of the world, or real systems, that we are interested in. It can be regarded as a meta-model over our modelling. An overlying concept is the first step towards a definition of both the model structure with its premises and the calculations possible to perform. In “Nordic Guidelines on Life Cycle Assessment” [Lindfors, 1995] a detailed descriptions of the steps involved in performing an LCA is described. It does not, however, give any information about data models or simulation procedures. The “Environmental life cycle assessment of products, Guide” and “Environmental life cycle assessment of products, Backgrounds” [Heijungs, 1992] also discuss the steps involved in detail. In addition it also propose two methods for solving the flow chart to the reference flow. The ISO 14041 recommendations [ISO 14041] discuss the general workflow for an LCA, but does only treat the subject on a general level. It does not, however, give any recommendations on how to calculate for a given problem. In the model area some promising data structures for LCI-data have been developed. Among them can be mentioned SPOLD [SPOLD] and SPINE [SPINE].

In this chapter the SPINE data structure is used to exemplify LCA modelling and simulation. The reason is there is no other LCA data structure available that is both consistent and flexible. It is also important to state what kind of data structure is used since it will give the premises for modelling using that particular structure. In most cases the examples in the chapter conform to what is commonly used in the LCA area.

3.1 Goal and scope

The goal and scope definition provides the application, the depth and the subject of the study (from [Heijungs, 1992] pp. 17). It includes a definition of the system and its boundaries and discusses why the study is carried out. Taking this into account it is clear that there is a close relation between the goal and scope and type of simulation. It does not, however, formulate what kind of simulations to be done. In fact it is not even mentioned that there might be several possibilities when it comes to simulation. The reason for this is probably that historically there is only one type of simulation that is performed: static solving of a linear model.

3.2 Model

Since dynamic simulation is not common most models are static and hence time independent. The most common type of model is the one shown in figure 6.1. This is a flow-oriented model where the connections between the gates in a block (in SPINE called activities), i.e. input and output terminals, are described with numbers. These given values are here denoted **u** for the inputs (inflows) and **y** for the outputs (outflows). The superscript shows which block they belong to; in the case of figure 3.1, block a. The correspondingly calculated values have an additional tilde.

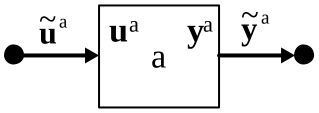

Figure 3.1. Static, linear and atomic or aggregated model building block used in LCI.

The given values can then be scaled linearly when the total model is evaluated. It is therefore useful to define a scaling variable as:

$$F_a = \frac{\tilde{\mathbf{u}}_i^a}{\mathbf{u}_i^a} = \frac{\tilde{\mathbf{y}}_i^a}{\mathbf{y}_i^a}$$

In addition to building blocks the model also contains connections between flows, which are called Flow Connections or '*fc*'. They can be used to connect inflows with outflows in the same or different blocks. In the case of more than one connection to a flow the ratio for each connection can be specified, see figure 3.2. The subscript '*or*' denotes Outflow Ratio and '*ir*' Inflow Ratio respectively.

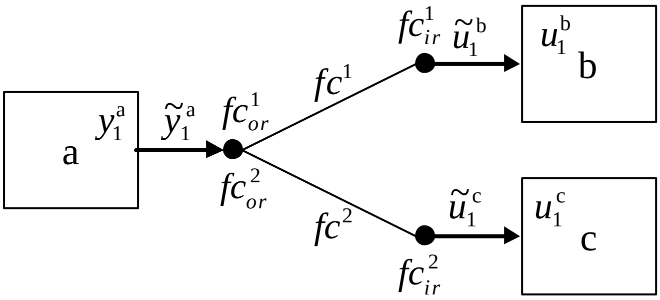

Figure 3.2. Flow connections between flows in a model.

This model is an a-causal formulation of a linear, static and non-equation formulated model. It means that no order of calculation is expressed in the model.

The aggregated block, i.e. containing other blocks are treated in the same way, except that its given values have to be calculated.

3.3 Simulation

Since the normal LCI-model is static, the simulations discussed here are all for static solving. In LCI it is called to normalise the system to the reference flow. The "Environmental life cycle assessment of products, Backgrounds October 1992" [Heijungs, 1992] suggest two alternatives for this on page 52: the sequential method and the matrix method. In the first method the chain of product demand is followed from the functional unit and upwards until all processes are and flows are calculated. One problem is that there is no easy way to solve the problem when there is a closed loop present. In the latter one, the matrix method, all data contained in the process tree is formed into a matrix. Then all flows are solved simultaneous by inverting the matrix. With this approach it is also possible to solve looped models. In this article Heijungs discuss software implementation difficulties with an upcoming LCI problem, e.g. a loop, without re-considering the interpretation of this loop, in terms of the mathematical aspects of the goal and scope and the actual inventory for the study. Such considerations and re-considerations are the aim of this work.

An interesting type of simulation performed in small scale in LCA is optimisation. In particular multi-objective optimisations have been successfully carried out on linear models using linear-programming (LP) algorithm [Azapagic, 1999].

Another method presented here is to solve the system for the amplification factors, F_i, for the processes first and then use these to calculate all flows. Doing so is a straightforward procedure where all equations given by the model are summarised and put into a matrix that is solved by using Gauss-elimination. In the process only the connected flows are evaluated. The reason is only to reduce computational complexity. Only the connected flows will contribute to the solution, the non-connected ones can be calculated using the activity amplification factor later.

The governing equations can be divided into the following categories:

1. Reference flow. Set the connected gate equal to the reference flow:
   $$\tilde{u}_i^x = x_{ref}$$
2. Activity internal connections. For each connected gate, relate it to the amplification factor as:
   $$\tilde{y}_i^x = F_x y_i^x$$
3. Flow connection mass balance. Formulate the mass balance equation for each flow connection node.
   $$\tilde{y}_i^x - \sum_i fc^i = 0$$
4. Flow connection ratios. In case of any constraints on the ratios for flow connections, formulate a relation for each node:
   $$\tilde{y}_i^x fc_{or}^i = fc^i$$

After the above equations are put together and sorted the equation system can be solved for the amplification factors. Then all flows are calculated by multiplying each of them with the amplification factor.

3.3.1 Loops

Loops are no problem, unless they are connected within one block, which could introduce an inconsistency to the problem, as in figure 3.3.

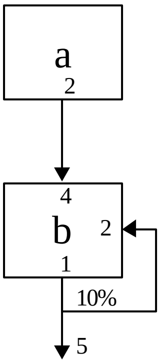

Figure 3.3. Inconsistent LCI flow model.

If the above equation system is evaluated we find that:

$$20F_b = F_b$$

The only solution is the trivial one, F_b=0. All other cases would lead to the impossible case 20=0. There is no solution to this situation and the model needs to be reformulated. The inconsistency is detected when the equations are gathered as an over-specified problem. If the model is changed to the one in figure 3.4 it can thus be solved, but only if the recycling rate is not fixed, but remained as a variable.

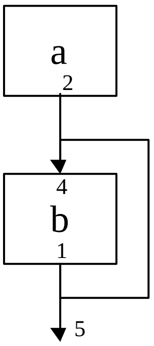

Figure 3.4. Solvable LCI flow model.

3.4 Future perspectives

The chapter ends by giving general suggestions on what can be done to extend the simulation capabilities in LCI and what kind of knowledge it would require.

The work can be divided into two parts: Extend the model description abilities and extend the number of available simulation for the model. Of course they are dependent of each other. Since the calculations in the simulation needs appropriate input data, the model has to contain them. The total dependency chain is shown in figure 3.5.

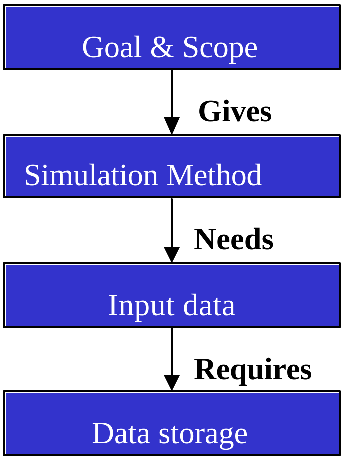

Figure 3.5. Data persistence dependency.

Therefore the first step would be to agree on the types of simulations one want to be able to perform. Next the requirements for the model can be found by evaluating the needs of these simulation types. Other requirements may also be include, such as possibilities to express non-linear elements etc. After all requirements are gathered one must agree on how to represent these in a mathematical and formalised way. This will lead to a syntactic definition for the model. Most possibly already available notations can be used, if not completely so at least partly. As indicated in appendix 1 the non-causal form is superior when it comes to extendibility, modularity, and separation of model and simulation as well as simulation types possible to perform.

The purpose of the project is to make use of knowledge of modelling and simulation from other engineering areas within LCI. To accomplish this a deep knowledge in the theories behind modelling and simulation is needed. Underlying concepts and ways of evaluating and structuring general modelling problems have to be applied on the special problem domains found in LCI. It is not the same as directly applying knowledge from other areas onto LCI. Since all applied areas are already specific, making use of the modelling knowledge directly would limit the usability of the results severely. Instead it is important to trace these specific approaches back to the general concepts and apply these on the problem. The proper role of related areas is to serve as idea generating examples of what can be done and as a source of inspiration. It can also provide usable experiences on both general and specific problems related to modelling. The area of application is here LCI. It is the reason behind the study and why it is performed at all. It is, nevertheless, not where the major effort needs to be put. The role of the LCI area is to provide the inputs for the modelling and simulation conceptualisation. Afterwards the result needs to be applied and interpreted in terms of LCI.

4 Conclusions

A number of techniques for performing an LCA have been investigated. They discuss the steps involved in performing an LCA into details. Just a few of these do, however, take the calculations performed in the simulation phase of LCI into consideration. Without exception these only can be used for linear and static problems, since no other element are allowed. LCI calculation can thus defined as to find a steady state solution for a linear problem. No source has been found where an underlying theory for simulation in LCI is presented.

From the large number of theories and methods available in other areas there is enough substance to provide a foundation to a general approach to modelling and simulation in LCA. No direct applicable theory has been found though. For the creation of such a theory recent important findings in control theory, such as non-causal models, can be applied. Additional structural information can without doubt be collected in the mathematical methods listed. Such information will be valuable for checking the models for logical/structural error in the early design process.

5 References

ASCEND IV, School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213-3891, ASCEND IV simulator, http://www.cs.cmu.edu/~ascend.

Azapagic, Adisa, 1999, “Life cycle assessment and its application to process selection, design and optimisation”, in “Chemical Engineering Journal”, 73 (1999), pages 1-21, Elsevier Science S.A.

Bossel, Hartmut, 1994, “Modelling and Simulation”, A K Peters Ltd.

Calahan, D.A, 1972, “Computer-Aided Network Design”, New York, NY: McGraw-Hill.

Carlson, Raul and Tillman, Anne-Marie and Steen, Bengt and Löfgren, Göran, 1998, “LCI Data Modelling and a Database Design”, Int. J. LCI, pp. 106-113.

Cellier, Francois E, 1991, “Continuous System Modelling”, Springer-Verlag.

Chua, L.O. and Lin, P.-M, 1975, “Computer-Aided Analysis of Electronic Circuits”, Englewood Cliffs, NJ: Prentice-Hall.

LCAiT, LCI inventory Tool, Stiftelsen Chalmers Industriteknik, Ekologik, Sweden, http://www.cit.chalmers.se/eng/site/frameset/program.htm.

Cytronix Ltd, 163 Summer St, Maynard, MA 01754, USA, http://www.cytronixltd.com/flowsheeting.htm.

DYMOLA, Dynasim AB, Research Park Ideon, SE-223 70 Lund, Sweden, http://www.dynasim.se.

GNU, GNU Electronic Design Automation, http://www.geda.seul.org/index.html.

Gross, Jonathan and Yellen, Jay, 1999, “Graph theory and its applications”, CRC Press.

Heijungs, R, 1992, Environmental life cycle assessment of products, Backgrounds – October 1992, Centrum voor Milieukunde, Leiden.

Heijungs, R, 1992, Environmental life cycle assessment of products, Guide – October 1992, Centrum voor Milieukunde, Leiden.

Heijungs, R and Frischknecht, R, 1998 , “A special view on the nature of the allocation problem”, International Journal of Life Cycle Assessment 3 (5) pp. 321-332.

ISO 14041, International Standard 14048:1998(E), Goal and scope definition and inventory analysis.

Karplus,Walter J, 1976, “The Spectrum of Mathematical Modelling and Systems Simulation”, Proceedings Eight AICA Congress on Simulation of Systems (L Dekker, ed), North-Holland, Amsterdam, pp. 5-13.

Korn,Granino A. and Wait, John V, 1978, “Digital Continuos System Simulation”, Prentice-Hall, Englewood Cliffs, N.J.

Levine, W.S., 1996, “Modelling from Physical Principles”, In “The Control Handbook”, ed. W.S. Levine, CRC Press, Boca Raton, FL, pp. 99-107.

LINK, LINK: A Combinatorics and Graph Theory Workbench for Applications and Research, Los Alamos National Laboratory, Computer Research http://www.c3.lanl.gov/~reid/Web_papers/genGraphToolTR/genGraphToolTR.html.

MATLAB, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA, www.mathworks.com.

MIMES, Energy Systems Technology, Chalmers University of Technology, SE-412 96 Göteborg, Sweden, The MIMES/Waste model, http://www.entek.chalmers.se/~josu/mimes.htm

Minsky, Marvin, 1965, "Models, Minds, Machines", Proceedings IFIP Congress, pp. 45-49.

MODELICA, Modelica: Language Design for Multi-Domain Modelling, http://www.modelica.org.

NIMBUS, Computer Aided Process Engineering Centre (the CAPE Centre), Department of Chemical Engineering, The University of Queensland, NIMBUS simulator, http://daisy.cheque.uq.edu.au/nimbus.

Lindfors, Lars-Gunnar et al, 1995, "Nordic Guidelines on Life-Cycle Assessment", Nordic Council of Ministers.

NP-Tools, Prover Technology, Sweden, http://www.prover.com

Rosenberg, Ronald C, 1983, "Introduction to Physical System Dynamics", McGraw-Hill.

SIMTECH, Abteilung für Simulationstechnik, Simulation Links, http://eurosim.tuwien.ac.at/hotlinks.

SPICE3, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA, http://infopad.eecs.berkeley.edu/~icdesign/SPICE/MANUALS/spice3.html.

SPINE data structure, CPM, Chalmers University of Technology, Gothenburg, Sweden, www.globalspine.com.

SPOLD data structure, SPOLD European Centre, att. Peter Hindle, Temselaan 100, B-1853 Strombeek-Bever, Belgium, www.spold.org.

Strömberg, Jan-Erik, 1994, "A mode switching modelling philosophy", Dissertation no. 353, Dept. of Electrical Engineering, Linköping University, Sweden.

Sundberg, J. and Wene, C-O, 1994, "Integrated Modelling of Material Flows and Energy Systems (Mimes)", International Journal of Energy Research, Vol. 18:3, p.359.

Wellstrand, P.E, 1979, "Physical System Modelling", Academic Press.

WWW Virtual Library, Electrical Engineering, http://arioch.gsfc.nasa.gov/wwwvl/ee.html#mfr.

WWW Virtual Library, Formal Methods, The WWW Virtual Library, Formal Methods, http://www.comlab.ox.ac.uk/archive/formal-methods.html.

Zeigler, Bernard P, 1976, "Theory of Modelling and Simulation", John Wiley, New York.

Appendix 1. Simulation properties

Here the theoretical part of simulation is described in more detail than in chapter 2 where the concepts were presented.

In the following an example with a space rocket will be used to explain the introduced concepts and how they relate to each other.

Example:
Consider a simulation of the vertical movement of a space rocket in different situations like starting from planets, travelling in space, landing on planets etc. (From [Cellier, 1991, pp 24 and 81]).

The rocket introduced above is here considered to be our system. We will, for simplicity, regard the rocket as a single piece, without any moving or detachable parts. Some literature suggests the specification of data inputs and outputs, i.e. to identify some cause-effect chains in the system. This approach is avoided here for reasons shown later. Instead we will identify the properties of interest or interface of the system. These properties can later be used as either data inputs, i.e. means of influencing the system, or data outputs, i.e. observations about the system depending on the type of experiment and settings. Observe that the notations input and output are here used for the flow of information only to show how data travels in the system. In this context these properties are the gravitation affecting the rocket (*g*) and the distance from the surface of the planet to the centre of gravity of the rocket (*x*), as indicated in figure A1.1.

Now we have defined two important issues on our system:

1. System boundaries.
2. Interface of the system.

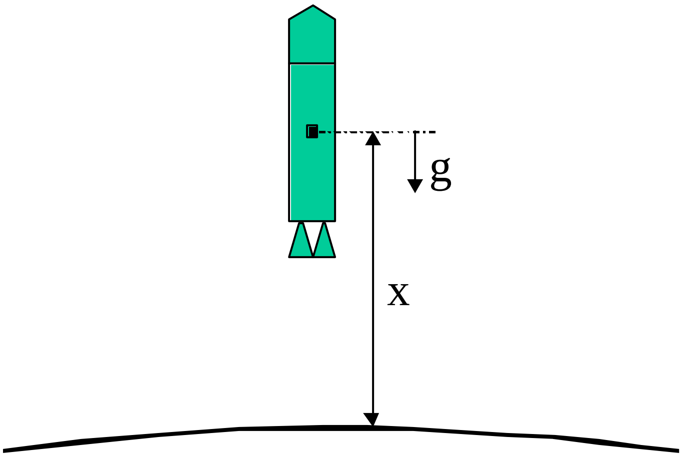

Figure A1.1. Example space rocket system with properties.

A1.1 Model types

There are two different main types of models: models in which signals are flowing and models in which, mostly physical, entities are flowing. The term flow is here used in its abstract form; no “real” flows need to be present. Instead it might be the weight force (*mg*) from a stone resting on the ground or an entropy flow resulting from a heated radiator in a cold room. To a certain extent a physical flow can be described with a single signal, so the real difference between these models lies elsewhere. It is a matter of how much information the model gives on how it is supposed to be calculated. In the first case the order in which the calculations shall be carried out is explicitly given. It is therefore the flow of information that is represented in the, often visually shown, graph model. In the latter case no inherent order of calculation is given. In fact the model does not give any information about calculations at all. Instead the flows represents a neutral connections transferring any entity, physical or not. The arrows are in this case only used for convenience, they make no difference on how the model is evaluated in a simulation.

Now a number of model types are presented. For each model type the example system from above will be modelled and advantages/disadvantages will be discussed shortly.

Example:
We will assume the case when our example rocket is starting from a planet and we want to know the current distance from the planet as a function of thrust and mass. The major physical law applied is Newton's law $F=ma$, where positive F is the total force pushing the rocket from the planet, that is $F=thrust-mg$, m is the momentary mass and a the momentary acceleration of the rocket, as shown in figure A1.2. Moreover the gravity constant g is said to follow $g=g_0/x^2$.

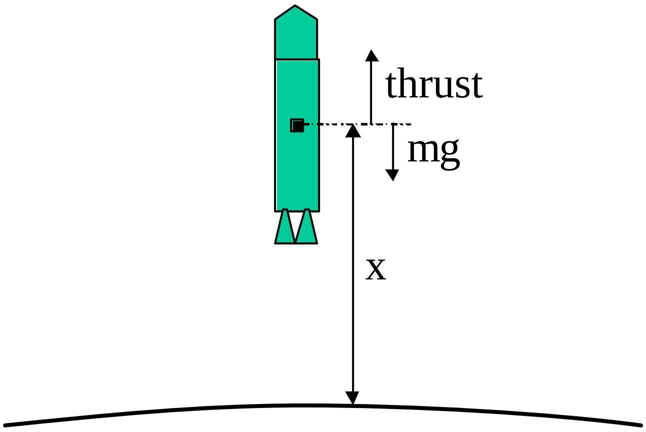

Figure A1.2. Forces applied on the rocket in the example.

In the following the equality sign (=) is used to indicate equality between two expressions without necessarily saying how to compute either of the two. The mere purpose is to state that they are equal. It does not suggest in what order any variable in the expression might be calculated. The assignment sign (:=) is used when an expression has to be calculated in a specific order. The variable to the left of the operator is calculated by evaluating the right side of the expression. In this case no other possibilities exist.

A1.1.1 Block diagram

The block diagram is a signal-oriented model in the sense that information travels or flows in a given direction only, denoted by the arrows. It is built from elements having pre-designated data inputs and outputs. In the time domain the output vector $\mathbf{y}$ is calculated from the input vector $\mathbf{u}$ using a transfer function, in this case g. For the frequency domain Y, U and G are used. The general visual representation is shown in figure A1.3.

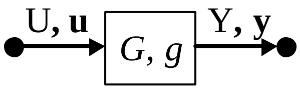

Figure A1.3. Graphical block diagram representation.

The output is then calculated as:

$$\mathbf{y}(t) := g(\mathbf{u}(t))$$

Usually linear models are expressed using the Laplace-transform, i.e. in the frequency domain. In this case the transformed variable, s, expresses the time varying properties of the model. When this type of representation is used the in and out-signals are normally scalars. Increased complexity of the model is expressed by extending the number of blocks. Due to the Laplace-transform the calculation of the output now becomes:

$$Y(s) := G(s) \cdot U(s)$$

An advantage with this type of model is the ease of understandability. Since the model is a graph it can easily be visualised, overviewed and understood. The model is also broken down into sub-models in a natural way due to the graphical layout. Important is also the large number of available tools in form of software to support block diagram models. The major disadvantage is the inflexibility of the types of simulations possible to perform on the model. It is a model where the calculation causality, or order, has been built in. It requires the modeller to decide on in what order the model shall be evaluated by the time of the design.

Example:
The law of Newton gives:

$$thrust - mg = thrust - \frac{mg_0}{x^2} = ma$$

Now a simple block diagram describing the process can be constructed directly from the formula above. Figure A1.4 shows one possible construction. The mass is for simplicity here taken to be constant, which of course, is not true in the real case.

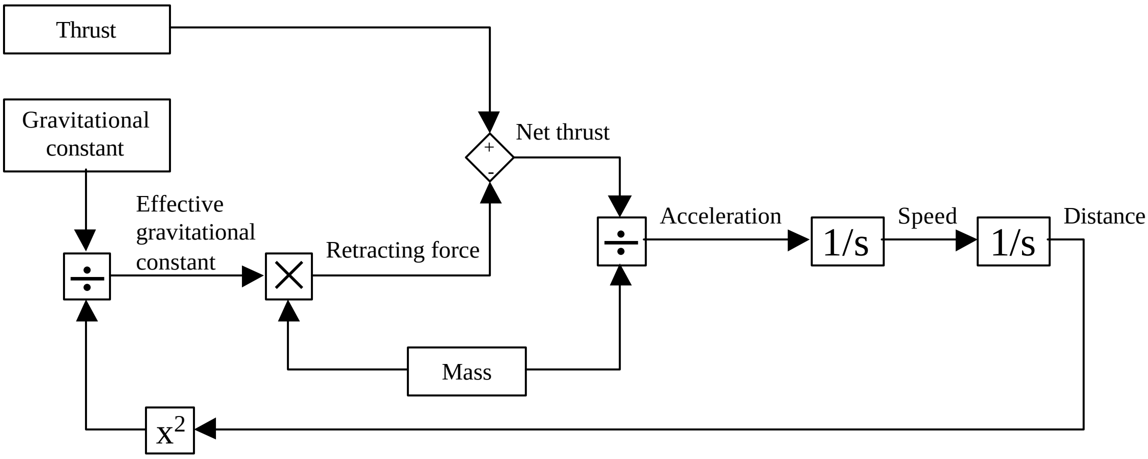

Figure A1.4. Block diagram over the rocket example.

A1.1.2 State-space models

This is the most commonly used form in control theory. It is a flexible way of describing both linear and non-linear multi input and output systems. It is also a form of signal oriented model where the calculations take pre-defined ways. The basic form is:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{u}, t)$$

The **x**-vector is the internal state vector, representing the “memory” of the model. The **u**-vector are the inputs and **y** the outputs. In the linear case it can be written as:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$
$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

A is called the system matrix, B the input vector and C the output vector.

The advantages are the compactness of the model description and in addition, this is an well-accepted form among control engineers. It is, however, not easy to understand the model by just examining the equation formulation. This form offers a more general representation than the block diagram. Since the model is formulated in terms of the derivatives it is somewhat more flexible for different kind of simulations.

Example:
The law of Newton is re-written in the form:

$$a := \frac{thrust}{m} + \frac{g_0}{x^2}$$

The internal states are chosen as the distance from the planet, here called x_1, and the velocity, here called x_2. The state-space model then becomes:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \frac{thrust}{m} - \frac{g_0}{x^2}$$

Since this is not a linear model it cannot be re-written in the somewhat more convenient way shown above. Instead it is used to derive the calculations desired. The inputs are then the total thrust, *thrust*, and the total mass, *m*. Output is the distance x_1.

A1.1.3 Equations models

This is a type of model with less formal requirements. In short all the equations, or relations, are formulated for a piece of or the entire model. It is in its theory based on equilibrium between variables and does not implicate any particular sequence of calculation. Even though it seems to be a trivial type of model it is very powerful. The advantages are that it is often easier to state the governing equations in a “raw” form that is closer to the physical law or phenomena behind the model. The total model is then built from many equations stating the constraints. The most important advantage is the ability to omit any pre-defined sequence of calculation, or calculation causality. The model can then be used for any type of simulation as long as the purpose behind building the model remains the same. On the other hand, the main disadvantage is the more complicated treatment of the system equations needed to be able to perform simulations. Presently there are not many existing tools to handle this kind of model formulation.

Example:
The equations behind the rocket example are now treated as expressions of equilibrium and are expressed in the version closest to its original, physical formulation. We now only have to state one equation, which is for the transversal movement:

$$thrust - \frac{mg_0}{x^2} = ma$$

A1.2 Simulation types

Below some basic simulation methods are discussed.

A1.2.1 Static solving

This is what we normally mean by "solving the system'". The model is evaluated for the steady-state case, i.e. if the model contains derivatives they are all set to zero. A static model can be evaluated directly. The reason for evaluating a dynamic model statically is that one wants to find the points of equilibrium, i.e. where the model is stable and no signal changes value.

The most commonly used mathematical algorithm in static solving is Gauss elimination. Depending on the form for the model, i.e. block diagram etc., different methods are used to gather and sort the equations involved. These equations are describing connections within the model, such as physical laws and structural elements, and all the constraints, such as different maximum and minimum values. They may also include some input to the model. Now they are all sorted and categorised as equations or unknown variables. In the case where the number of equations matches the number of variables the problem is said to be well posed and can be solved. This will give us a square system matrix. When the number of equations is larger than the number of variables the problem is over-specified. That is because too many constraints have been put on the problem. No exact solution can be found under these conditions, only an approximate one is reached. An approximate solution will not exactly satisfy all the conditions or constraints put, but when calculated using the least square method the nearest possible value for all variables are found. On the contrary, when the number of variables is more than the number of equations the problem is under-specified. This is a result when we have not specified enough constraints on the problem to fix it to one solution only. Now any number of solutions can be found that satisfies all the constraints. The solutions are said to occupy a space. The number of dimensions in this space of solutions is calculated as the number of variables minus the number of equations.

A1.2.2 Dynamic solving

Dynamic solving or dynamic simulation is to evaluate a model and record its response over a period of time. This only makes sense when the model includes any dynamic elements, resulting in different types of derivatives in the equations. Before the actual dynamic simulation is started the model is sometimes solved statically for a defined set of inputs. Such an evaluation will ensure the initial state to be a state of equilibrium for the set of inputs, i.e. set the rest of the variables accordingly. When the dynamic simulation starts, new dynamic sets of inputs are then fed to the calculation that will make the model respond. Another case is when the initial state is not in equilibrium and the simulation aims at finding what will happen when the system is released from such a state. No new inputs are fed to the calculations and the responses will either, in case of dissipative elements present, decrease until finally a point of static equilibrium is reached or, in case of no dissipative elements present, stabilise at a certain frequency and amplitude.

Calculating the derivatives for the present state and integrating these over a step in time to reach the next state usually performs the dynamic solving. Each step in time will produce a new set of variables, so the total amount of data is often large. If the model contains derivatives that change slowly over time i.e. long time constants, the time step can be lengthened.

A1.2.3 Optimisation

Optimisation is to evaluate the model for a number of different inputs, or combinations of inputs, to find the optimum case. To be able to determine this a target function is formulated. The target function can be anything from just a variable to a mathematical function calculated from a number of variables. In the optimisation either a maximum or a minimum for the target function is found by changing the variables. Since most variables cannot be extended infinitely some constraints are put for each involved variable. The target function always evaluates to one number only, even if it can be calculated using many variables. The number of variables can be more than one, in which case it is a multi-variable optimisation.

To perform an optimisation one needs an Under-specified problem, i.e. a problem where one or more variables are allowed to vary and a (static) solution is still possible. Then a different optimisation method or algorithm is used depending on the type of model. In the case of a linear model a Linear Programming (LP) algorithm may be used while in non-linear case a non-linear programming (NLP). For the non-linear case it can be difficult to determine if the optimum found is a global one, i.e. an optimal point for the entire defined interval and not only a local one. This is the main issue of the area Global Optimisation.

A1.3 Modelling techniques

The modelling technique is different depending how much and what one knows about the system. A well-known system may be broken down in its structure and each part modelled using physical laws or other mathematical forms. A less well known system may require sampling of the real behaviour and formulation of an approximate mathematical form with very limited validity, a so-called identification. The former is usually referred to as a “black box” model and the latter as a “white box” model. In between these extremes there is a spectrum of different “gray box” models, depending of the degree of inductivity. [Karplus, 1976] It is practical to describe the model differently mathematically depending on how much one knows about the underlying phenomena. Even though all systems exist in the continuos time domain we can still have discretion of part of the description when we do not know enough. Of course this will limit the possibilities to perform some sort of simulations on the model.

A1.4 Purpose

The purpose of the study is the very first priority to carefully consider when to perform a simulation. It is important because it will give shape of the premises and limitations for all steps involved in the process. Moreover, it is the link between the model and the simulations performed on the model. In contrast to the system, which represents a piece of reality and thus is valid for all experiments, the model is created for a certain purpose and a certain range of simulations. Outside this range the model is no longer valid. This becomes very clear of the fact that a model is a simplification of reality made in order to answer one or a set of specific questions.

In the literature, e.g. [Cellier, 1991], it is often argued that the experiment, or in this case the simulation, is connected to the model. They should therefore be treated together in a so-called *system and experiment* tuple. Other sources suggest the introduction of an *experimental frame* [Zeigler, 1976] to relate the model to a specific set of simulations. The dependencies of these can be shown in a simplified form as in figure A1.5. Here activities, i.e. active processes, are symbolised by diamonds and resulting passive documents by rectangles. The activity “Simulation Method Creation” can be substituted for finding the right method, if such a method already exists. All activities are either directly or indirectly dependent on the purpose of the study. It is obvious that some correlation between model and simulation method is needed. On the abstract level it is the purpose of the study, but a more concrete and logical connection is needed for proper verification and validation.

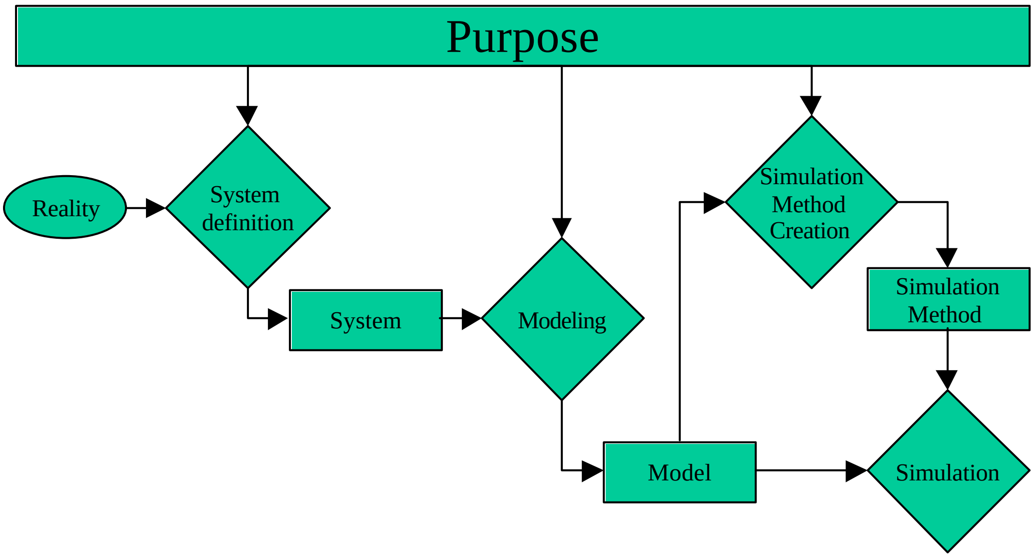

Figure A1.5. Connections between activities involved in modelling and simulation.

Example:
For our space rocket example we need to build a very general model, which will be valid for a number of different simulations stated above, such as starting from planets, travelling in space, landing on planets etc. Specifically we are interested in calculating the behaviour when starting and taking off from a planet. This is the purpose of our study.

A1.5 Causality

Causality is a controversial subject when discussing simulation and model building. Some people argue that reality is causal and that it is a requirement to be able to perform simulations on it. Others claim that the cornerstone in reality is energy storage and equilibrium [Levine, 1996, pp. 99-107]. A closer examination shows that there are a number of different causalities [Strömberg, 1994, ch 1]. In addition to the physical causality, or the causality of nature, there is also a computational causality that determines the calculation order of the equations. In the first case it is the inherent property of nature to react to stimuli that cause an action. It is evident that the nature is following an action-reaction pattern. In this way most of the phenomena of nature can be structured in action-reaction chains. These chains can then be connected to form large structures that can be used to mimic the behaviour of complex systems. This kind of model works well for the purposes they were developed for but is usually hard to re-use for other. The reason is they contain information of how the model is to be used, which leads to a certain order of calculation. This order of calculation is called computational causality. For a causal model a certain variable, or property, is calculated. In case we want to investigate some other property, then we have to rebuild the model. In the case of large, complex system it is not always easy to do so. If we, however, remove any assumptions on the calculation order and only state the governing equations, it is possible to re-use the model for other purposes. These models are called a-causal. Whatever the real truth is about causality in nature it can be readily shown that the removal of causality from the model is beneficial in many ways. It will in addition make the model more modular [Strömberg, 1994, sec 1.2]. The drawback is that to be able to make calculations, there has to be a calculation order. If it is not given explicitly in the model it has to be added later, either manually or automatically.

A1.6 Closed loops

Sometimes a closed loop can be a problem. Some states it is always a problem, even impossible to solve. This part gives some reflections of the nature of closed loops.

A closed loop is some sort of feedback to “earlier parts” of a system or model. The definition of an earlier part differs depending on if it is a real system, causal or a-causal model, and so does the implications. In the case of a real system it usually means some sort of recycling where something is looped back to earlier stages in the production line. This is no problem and poses no difficulties in our real world. When a model is build to reflect loops it might, however, lead to inconsistencies. In some cases it is due to the modelling concept used, i.e. the possibilities to describe the structure, and in other it is due to an erroneous model that does not reflect the real system in the proper way.

For the causal model, where the calculation sequence is fixed, a closed loop means a mathematical feedback to earlier stages in the computational chain. If using only direct feed-through[1] components it means that the input to calculate a result is dependent of the result itself [Bossel, 1994, sec 3.1.3]. It is then called an algebraic loop. A simple example is shown in figure A1.6 below. Here the output, y, has become a function of itself:

$$y = K1 * u - K1 * K2 * y$$

For a model where the calculation order is fix, this is a problem. Such problems can be solved by iteration, using in the case of MATLAB [MATLAB], a Newton-Raphson technique. Solved analytically we find that:

$$y = \frac{u}{2}$$

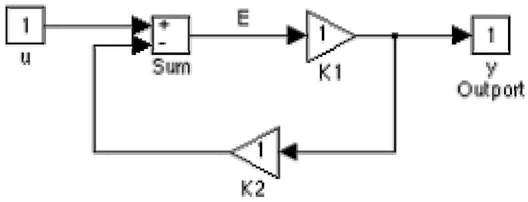

Figure A1.6. Algebraic loop using SIMULINK block diagram in MATLAB.

For the a-causal model a loop is like any other connections between two ports. It represents a constraint that can be added to the list of governing equations. As long as the loop is not inconsistent it does not give any problems.

Inconsistent loops occur when the simulation equations have only trivial solutions; i.e. the only solution is a zero flow trough the circuit. Depending on the model concept it can arise for different structures.

---

[1] A direct feed-through component is a component that immediately in time calculates the result from the input(s) and transfers it to its output(s). In other words it is a static component without a memory.

Appendix 2. Survey and categorisation

Many related engineering areas derive models of technical systems and use advanced calculation methods on them. Most of these are, however, specialised for a specific purpose and cannot directly be used within LCI. On the other hand, both the outlines of the method and experiences from using it can be re-used. That happens when one wants to transform and utilise them in another field, in this case LCI. Though up till now only few of such efforts have been made.

This is a general survey of present technology on physical flow modelling and calculation. It emerges from the need to find a uniform way to describe and perform calculations on general physical, multi domain systems within LCI. From the present vast number of techniques and approaches it can only present a few, selected by their consistency with underlying physical principles and their generality. The list is by no means complete and can be changed in the future. Even though the type of results, etc are also of interest, a solid foundation for the theory is needed to make it re-usable and extendable within other areas.

In the first version all searches have been performed through Internet with public access tools, which makes the number of computer application tools dominate. In order to make it complete a literature search needs to be continued.

A2.1 Categorisation

The categorisation of techniques and tools is based on what is described above. Since different areas, i.e. methods and software tools, are different and thus different types of categorisation are needed, two separate types are developed. In addition a short general description is given.

A2.1.1 Simulation Methods

Simulation methods are first categorised according to the purposes they can be used for. Such purposes may include static solving, dynamic simulation etc. Also the used algorithm or algorithms are given, if it is possible to retrieve these. Next the properties of the model are investigated. This includes linearity, time and space dependencies and causality in description.

Simulation	Purpose	Stationary solution, Dynamic simulation, Optimisation, Comparison, Uncertainty analysis etc.
	Algorithm	Gauss elimination, Linear programming etc.
Model	Linearity	Linear – Non-linear
	Time	Constant – Discrete – Continuous
	Space	Constant – Discrete – Continuous
	Causality	Causal description – Non causal description

Table A2.1. Categorisation criteria for simulation methods.

A2.1.2 Simulation software

Simulation methods are categorised using the same criteria as methods and in addition also some on modelling approach and storage type. The modelling approach is divided into a spectrum from purely inductive to purely deductive. The storage structure, or design, is considered as well as the media.

Simulation	Purpose	Stationary solution, Dynamic simulation, Optimisation, Comparison, Uncertainty analysis etc.
	Algorithm	Gauss elimination, Linear programming etc.
Model	Linearity	Linear – Non-linear
	Time	Constant – Discrete – Continuous
	Space	Constant – Discrete – Continuous
	Causality	Causal description – Non causal description
Modelling	Approach	Inductive (Black Box) – Opaque (Gray Box) – Deductive (White Box)
I/O	Structure	SPINE, SPOLD, MODELICA etc.
	Media	File, database, internet, intranet, organisation information system, etc.

Table A2.2. Categorisation criteria for simulation software.

A2.2 Survey result

A2.2.1 LCI-area

Simulation Methods

Heijungs, et al

Category

Simulation	Purpose	Stationary solution
	Algorithm	Gauss elimination
Model	Linearity	Linear
	Time	Constant
	Space	Constant
	Causality	Non causal description

Description

Describes the technical system as a characteristic matrix, which defines all flows for the processes included [Heijungs, 1998]. Only static functions describing non-aggregated systems are allowed. The functional unit is expressed as a net flow vector and the result, here called inventory vector, can be calculated by standard matrix inversion. Parts of this approach come from economical models where it is used to calculate production demand, whereas here in this context it is tailored to fit in to LCI.

Simulation software

LCIiT 4.0

Category

Simulation	Purpose	Stationary solution
	Algorithm	Gauss elimination
Model	Linearity	Linear
	Time	Constant
	Space	Constant
	Causality	Non causal description
Modelling	Approach	Inductive (Black Box) – Opaque (Gray Box) – Deductive (White Box) (All modelling approaches supported through recursive definition)
I/O	Structure	SPINE + additions for calculation
	Media	File (xfr data transfer) and database (Access)

Description

State-of-the-art LCI calculation software that deals with static, linear models. Can do all kind of loops, as long as they are not inconsistent. Solves the given model statically under the constraints given [LCIiT].

A2.2.2 Related areas

Simulation Methods

Chemical Engineering

Process Flow-sheeting (Process Modelling)

Category

Simulation	Purpose	Stationary solution, Dynamic simulation
	Algorithm	Not given
Model	Linearity	Non-linear
	Time	Continuous
	Space	Constant
	Causality	Causal description

Description

Developed for modelling and simulation of chemical reactions in process plants. Uses flow sheets to describe and visualise processes to be calculated and simulated, often dynamically. Requires a fundamental knowledge about the process to describe the underlying equations. The system under calculation is then transferred to mathematical form and the solution(s) calculated. The method is stringent in using physical constraints such as mass balance, physical unit match etc. Normal output is static and dynamical behaviour. It is used not only to provide answers to both dynamically changes such as start and stop of process plants, but also to estimate the result of different changes in processes.

A number of software packages are available that performs both modelling and calculation/simulation. The Nimbus simulator [NIMBUS] and the ASCEND simulator [ASCEND IV] are both targeted at process modelling and simulation. Some attempts have been made to extend flow-sheeting to be used in biological processes [Cytronix Ltd].

Electrical Engineering

Energy Systems

Category

Simulation	Purpose	Dynamic simulation, Optimisation
	Algorithm	Non-Linear programming and more
Model	Linearity	Non-linear
	Time	Continuous
	Space	Constant
	Causality	Causal description

Description

Used to model and calculate energy systems that can be described by materials and energy balances but where the details of phase transition and chemical reactions can be left out. Both dynamical simulation and optimisations can normally be performed [Sundberg, 1994]. Most models are restricted to using only energy and material flows, even if the optimising function often is expressed in economical terms. The technique is also used to model waste treatment [MIMES].

Electric Network Analysis

Category

Simulation	Purpose	Stationary solution, Dynamic simulation, Optimisation, Comparison, Uncertainty analysis and more
	Algorithm	Not given
Model	Linearity	Non-linear
	Time	Continuous
	Space	Constant
	Causality	Non causal description

Description

Classical method for electrical circuit calculation and simulation [Chua, 1975][Calahan, 1972]. Only applicable to systems that can be expressed in terms of their electrical equivalents, i.e. different types of voltage-current relations. Deals with static, dynamical and optimising computations. Also different sensitivity studies, mostly in the case of temperature changes, can be performed. Major computer applications include different types of SPICE [SPICE3], even though a large number of both other non-commercial [GNU] and commercial [WWW Virtual Library, Electrical Engineering] alternatives exist.

Physical System Modelling (Control Systems)

Category

Simulation	Purpose	Stationary solution, Dynamic simulation, Optimisation, Comparison, Uncertainty analysis etc.
	Algorithm	Not given
Model	Linearity	Non-linear
	Time	Discrete – Continuous, depending on system
	Space	Discrete – Continuous, depending on system
	Causality	Both Causal descriptions and Non causal descriptions exist

Description

General method to model any kind of physical system [Rosenberg, 1983][Wellstrand, 1979]. Transfers the problem directly to control systems mathematical descriptions, where the whole set of calculations available in control system theory can be applied. This include dynamical simulation, steady state calculation, sensitivity analysis and in addition a vast number of description of the dynamic behaviour of the system under study. Mostly used to simulate or sometimes emulate systems together with a control device to study their behaviour. Today a large number of software tools has been developed for system control and signal processing applications [SIMTECH].

Multi Domain Analysis (Modelica)

Category

Simulation	Purpose	Stationary solution, Dynamic simulation, Optimisation, Comparison, Uncertainty analysis etc.
	Algorithm	Not given
Model	Linearity	Non-linear
	Time	Discrete – Continuous, depending on system
	Space	Usually Constant
	Causality	Non causal description

Description

First approach to describe multi-domain physical systems [MODELICA]. Modelica is a state-of-the-art standardised external model representation developed to solve the interoperability problem amongst the large variety of modelling and simulation environments available today. The board consists of a large number of leading research organisations, universities as well as commercial companies. Most probably more and more simulation tools will support this description in the future.

Computer Engineering - Mathematics

Graph Theory

Category

Cannot be categorised using the above scheme. Deals with structures and combinations in the structure.

Description

A method to analyse structures expressed in mathematical graphs, i.e. a number of nodes connected with each other to give a structure [Gross, 1999]. Traditionally used for evaluation of problems in discrete mathematics and combinatorics. The classical question to provide answers is “What is the shortest way between point x and y, provided that each node must be visited only once?” Also a number of other structural answers can be found. One computer application has been found that supports experimenting with graphs and graph theory related problems in an open and non complicated way [LINK].

Formal Methods

Category

Cannot be categorised using the above scheme. Deals with the logic of structures.

Description

A way to describe structures in a logical manner [WWW Virtual Library, Formal Methods]. Used to verify computer program, decision making chains, building construction checking, behaviour conformance checking etc. It is constantly finding new areas of usage due to its versatility. The output provides information about logically consistency in different aspects. Mostly used computer tool is NP-Tool [NP-Tools].

ASCEND

Category

Simulation	Purpose	Stationary solution, Dynamic simulation and Optimisation. More modules can be connected.
	Algorithm	Slv, QRSlv, LSODE, MINOS, LSSlv, Opt (SQP), CONOPT, Make MPS etc. Depending on simulation.
Model	Linearity	Non-linear
	Time	Discrete – Continuous (ODE and DAE)
	Space	Discrete – Continuous
	Causality	Non causal object oriented description
Modelling	Approach	Inductive (Black Box) – Opaque (Gray Box) – Deductive (White Box) (No restrictions)
I/O	Structure	ASCEND IV
	Media	File

Description

The ASCEND simulator [ASCEND IV] is an interactive, equation-based simulation application targeted at process modelling and simulation. It has an open structure that allows any system that can be described by differential and algebraic equations to be simulated. The open structure makes it possible to attain user defined analyses, such as error propagation etc through user supported mathematical modules.

SPICE

Category

Simulation	Purpose	Stationary solution, Dynamic simulation, Sensitivity analysis and Optimisation
	Algorithm	Not available
Model	Linearity	Non-linear
	Time	Discrete – Continuous (Both available)
	Space	Constant
	Causality	Non causal description
Modelling	Approach	Deductive (White Box)
I/O	Structure	SPICE
	Media	File

Description

SPICE [SPICE3] is a general-purpose analogue electrical circuit simulation program for non-linear dc, non-linear transient, and linear ac analyses. Only electrical circuit elements are available. Other types of systems need to be translated to their electrical counterpart to be modelled in SPICE.

Dymola

Category

Simulation	Purpose	Stationary solution, Dynamic simulation
	Algorithm	DASSL, LSODE, DOPRI, DEABM etc.
Model	Linearity	Non-linear
	Time	Discrete – Continuous (ODE and DAE) (Handles time- and state events)
	Space	Constant
	Causality	Non causal object oriented description
Modelling	Approach	Inductive (Black Box) – Opaque (Gray Box) – Deductive (White Box) (No limitations)
I/O	Structure	DYMOLA and MODELICA
	Media	File

Description

Dymola [DYMOLA] provides an object oriented modelling and simulation environment for model construction, simulation and graphical representation. The models are described in a non-causal equation oriented way. Object orientation enhances the modularity of the models. The simulation module handles ordinary differential equations (ODE) and differential-algebraic equations (DAE). It has a variety of numerical integration methods: one-step, multi-step and extrapolation methods. Mostly used for dynamical simulation of large hybrid systems, such as real time robot control, power plant simulation etc.