MODELLER—AN INTERACTIVE MODEL EDITOR FOR PHYSICAL-CHEMICAL-BIOLOGICAL PROCESS MODELS

Heinz A. Preisig and Mathieu R. Westerweele
TU Eindhoven
5600 MB Eindhoven, The Netherlands
H.Preisig@tue.nl

Abstract

Modeller is a tool that implements a systematic approach to model design. It builds on network modelling of physical-chemical-biological processes and currently supports lumped system dynamics, order-of-magnitude assumption handling, DAE-index reduction and instantiation of simulation models generating MatLab® files for immediate solving with its DAE-1 solvers.

Keywords

Computer-aided modelling, network models, model reduction, index reduction

Introduction

A chemical engineer is often asked to describe the dynamic and/or static behaviour of a physical-chemical-biological (PCB) process, because information about this behaviour is needed for analysis, control, design, simulation, optimisation or process operation. The mathematical model, which requires the use of all the basic principles of chemical engineering science, such as thermodynamics, kinetics, transport phenomena, etc., is the key to many chemical engineering problems (e.g. Cellier, 1991; Marquardt, 1994; Ogunnaike, 1994; Hangos and Cameron, 2001; Perkins et al, 1996). It should therefore be approached with care and thoughtfulness (Stephanopoulos 1984).

Modelling: An Integral Part of Problem Solving

The main objective of a mathematical model is to describe some behavioural aspects of the modelled process (Aris, 1978; Denn, 1986; Ogunnaike, 1994). The modelling activity should not be considered separately but as an integrated part of a problem solving activity. Preisig (1991b, 1991a) analysed and decomposed the overall task of problem solving into the following set of subtasks:

(Primary) Modelling. The first step in the process of obtaining a process model is the mapping of the "real-world" object into a mathematical object, called the *primary model*. In doing so, one may take different views and accordingly apply different theories, which naturally will result in different models. Within this first step, assumptions are made about the principle nature of the process such as time scales of hydraulics and reactions, fundamental states, etc.

Model manipulation. The model can be simplified by applying mathematical manipulations, such as: (i) Model reduction, (ii) linearisation, (iii) transformation to alternative representations of the model and (iv) re-arrangement of the mathematical problem equations.

Problem specification. A mathematical problem is specified by instantiating a consistent set of variables (i.e. defined as known), such that defined problem is structurally solvable.

Analysis of the mathematical problem. This analysis is done in conjunction with the problem specification. It includes a degree of freedom analysis, an analysis of the bipartite equation graph for structural solvability and the index of the differential algebraic system to be one or zero.

Solution of the mathematical problem. General-purpose mathematical packages, such as differential algebraic equation solvers, large-scale simulators, linear algebra packages, etc., are used to solve the specified problem.

Analysis of the solution. This analysis must focus on a verification of the results by comparing them with known plant facts such as experimental data or just experience.

Each of the above tasks must meet a set of specifications otherwise its, or one or several of the previous tasks', conditions must be adjusted by looping back to the appropriate task in the sequence. This implies that model design is a recursive and iterative process, which includes all aspects of the modelling process with a view on the intended use of the model. Rarely does one in the first attempt obtain a model that is appropriate for the intended use, but an adequate model is constructed progressively through nested loops comprising a series of tasks of model development, model validation and model utilisation.

Computer-Aided Modelling Goals

More often than not, the time spent on collecting the information necessary to design an adequate model is much greater than the time spent by the solver. Most publications and textbooks present the model equations as a given fact without mentioning or discussing the quite extensive underlying set of assumptions, which makes it obviously difficult to grasp the information contents of a model. Even more, it is considered good practice to represent the model in the measured quantities preferably in the form of ordinary or partial differential equations thereby eliminating all state variable transformations and other algebraic components. The derivation process yielding such models hides most assumptions thereby inhibiting any critical evaluation of the model and the assumed conditions. It makes "modelling" an unstructured process thus more of an art than a science asking novice modellers to learn dynamic model development by studying published examples built by more skilled modellers in textbooks and publications, or learn painfully through trial and error (Moe 1995).

Restructuring of industry, increasing demands on process predictability for the purpose of design, scale-up, improved and tighter or optimised control have dramatically increased the demand for process models. Industry and science asks for a multitude of different models, which depend on the intended use of the model: Models that cover different time-scale windows, different levels of detail, different underlying conditions reflecting into different assumptions and different performance requirements all together possibly asking for a vast number of different models for one and the same process. At the same time, models become increasingly more complex, which makes the model construction process even more time consuming and error-prone (MacKenzie and Ponton, 1991, Marquardt, 1991, 1992). All this calls for a systematisation of the modelling process and thus the development of an appropriate, well-structured modelling methodology for the efficient design of adequate, sound and consistent process models. Modelling tools building on such a systematic approach must support teamwork, re-use of models, provide complete and consistent documentation and, not at least, improve process understanding and provide a foundation for the education of process technology (Bieszczad, 2000; Moe, 1995; Preisig et al., 1991). It is the objective of this project to provide a systematic model design method that meets all mentioned requirements and turns the art of modelling into a science of model design.

Structural Elements

Networks

The models underlying the modelling program MODELLER are network models with two main elements, namely the nodes representing capacities and the connections (arcs) representing the flows of extensive quantities between capacities. A **System** we simply define as *part of the universe that shows capacity effects* implying the ability of storing extensive quantity such as mass, volume, energy and momentum and a **connection** as *coupling of two systems that shows no capacity effects and has resistance to the flow of extensive quantity*. In the limited, but most common application, a system represent a piece of the volume occupied by the modelled process and a simple system is then *a body of finite or differentially small volume consisting of a single phase or a pseudo-phase*. The connections are a representation of the abstract boundary separating two adjacent systems. Along the lines of classical thermodynamics connections are typed with the kind of primary extensive quantity being transferred. Thus we define connections for (component) mass (with inferred flow of volume work and total energy), heat and work.

Physical Topology

The network of capacities, which for the limited class of energy and mass models are frequently called control volumes, represent the physical containment of the plant, for which reason we named it *physical topology*. This physical topology may be hierarchically structured by utilising the general definition of system, which allows each system to be a network of internal connections and subsystems. With this definition, models become trees with the root being the process itself and the leaves being simple systems. The description of the whole process is then the collection of the leave nodes with their network of connec-

tions. The advantage of this representation is obvious as it allows for the representation of arbitrary complex systems.

Species Topology

The species topology defines which species are present in what part of the process and thus defines the basics of the chemistry/biology of a processing plant model. The species topology is a colouring of the physical topology. Species are injected in particular places and propagate through the network utilizing the paths opened by the mass connections. The mass connections may be selective in that a permeability is defined, which either allows a species to pass or not (Preisig, 1994). New species may be generated through reactions that are enabled whenever a set of reactants is present. Thus when defining the species topologies, the user defines a plant species set and a plant reaction set. The reactions are associated with capacities, thus systems and species are injected into particular capacities. The MODELLER then generates the species topology automatically. It also supports all possible modifications, including the modification of the physical topology, the location of the injection point, presence or absence of reactions, permeabilities and other structural changes that may be made to the model in the course of its modification.

Equation and Variable Topology

Once the species topology is imposed on the physical topology, the dynamic conservation equations can be written. The conserved quantities, namely species mass and total energy, are the fundamental extensive quantities that describe a mass-energy system with fast momentum transfer. The dynamic equations for the network, which form the core of the representation, can be written in compact form (Westerweele et al., 2000):

$$\frac{d\,x}{d\,t} = F\,z + R\,r$$

with x :: the conserved fundamental quantities, representing the primary state, t :: time, z :: the flows (of extensive quantities) and r :: the transformation of extensive quantities (reactions). The quantities x, z and r are stacks of vectors. In the case of the primary state it is the stack of state vectors, one for each system. The flow vector is the stack of all flow vectors and r is the stack of all normed reaction rates. The matrix F is the (typed) flow matrix representing the directed graph of the physical topology. The R matrix is a block matrix with the conversion relations for the various species, the stoichiometric coefficients, a concept that can readily be extended to species in different states of aggregation or other modified conditions, such as part of a complex, adsorbed etc. We also derived a procedure, which when applied to the above equation yields a minimal representation limiting particular the presence of species to where they are actually modelled to be present through injection, reaction or transfer. Details on this can be found in (Westerweele, 2002).

The next step is to link the flow and reaction variables to the primary state variables if they are not simply defined as part of the problem statement. The linking requires the definition (selection) of a transfer law, which defines in turn new variables, such as the potential or the properties of the *thermodynamic wall* being represented by the connection. The potentials we classify as secondary states, which can be linked to the primary state. In the case of *temperature*, to mention an example, it can be linked to the primary state *energy* through Legendre transformations. If the resulting set of algebraic equations is lower triagonal and with the secondary states completely linked back to the fundamental state, we call the model **proper**. For these models the set of variables that must be defined to completely solve a simulation problem can be identified through a bipartite graph analysis of the equation set (Duff et al., 1986; Elmqvist et al., 1995). This analysis is built into MODELLER and is engaged whenever the user wishes to instantiate a simulation problem.

Adding Assumptions

The **first set of assumptions** is made when defining the physical topology. By its definition, the user implies a window in the time scale, which is spanned by the smallest and the largest time constant defined in the model. These time constants are not known at this point in time. It is the user who maps his view of the process into an initial physical topology.

A **second set of assumptions** usually enters once a partial proper model has been defined in the form of order-of-magnitude assumptions in which the dynamic window is narrowed by assuming fast systems that reach their final state extremely rapidly. Similarly one may assume connections and/or reactions to be fast. For example a heat flow or mass flows that is much faster than the rest of the dynamically relevant parts of the plant. These flows can be eliminated through combining the connected system and defining an equilibrium relation reflecting the steady state relation between the two states connected through the fast connection. Similarly for reaction, if a reaction is assumed very fast, an equilibrium relation provides the necessary additional information. Both cases generate index problems. We have developed methods and procedures to resolve all index problems locally as they occur during the definition process (Westerweele et al., 2001). These procedures are implemented in the current version of the MODELLER.

Some Comments to the Implementation

The MODELLER is designed to effectively assist a model designer with the construction of consistent process models and to seriously reduce the needed time effort. All performed actions can be undone (multiple undo/redo mechanism) and the software allows to store, retrieve, import or export models (or parts of models) at any stage of the

model definition. This allows for a safe mechanism of model reuse and inheritance.

MODELLER is a context specific graphical editor, which supports the construction and modification of **hierarchal physical topologies**. The model window shows always the complete plant zoomed into a particular system, which is shown as a full network. The rest of the plant is shown on its highest possible node, relative to the zoomed node.

The **species topology** is computed by first finding the mass transfer networks and identifying the species present in each mass transfer network. A colouring algorithm then simply works through the network starting at the injection points and terminating whenever all injected species have been processed. In each primitive system the program checks if a reaction is enabled after a new species has been added. If the reaction is enabled, the products are added to the list. The new species are injected into the current system and the algorithm proceeds. The result is the multi-coloured physical topology.

Simulation Model Instantiation

The MODELLER implements the instantiation of simulation models, in which, roughly speaking, the initial conditions and all parameters as well as all controller setpoints and trajectories are defined. In this procedure the user may implement **order-of-magnitude assumptions** by choosing, for example, a system to be very small and thus very fast, indicating the feasibility of a pseudo steady state assumption for the corresponding part. He may also identify fast transfers between systems for which he must either assume a fast connection and define an equilibrium relation as a substitute for the transfer equation or assume steady state for one of the connected systems. At this point, index problems are detected and locally resolved.

Conclusions

The MODELLER III implements

- network modelling of general mass-energy processes;
- definition of proper, structurally consistent models in a quick manner;
- instantiation support for simulation models enabling order-of-magnitude assumption and automatic model reduction yielding always DAE 1 models;
- generates output that can be directly plugged into MatLab® to be solved by a DAE solver.

MODELLER III is the first of its kind. For the next version, we plan the extension to proper distributed models and different output languages enabling the link to other packages. Further, the instantiation component shall be extended to accommodate consistent instantiation of design problems.

References

Aris, R. (1978). *Mathematical modelling techniques*. Pitman, London.

Bieszczad, Jerry (2000). *A Framework for the Language and Logic of Computer-Aided Phenomena-Based Process Modeling*. PhD thesis. Massachusetts Institute of Technology. Massachusetts, USA.

Cellier, Francois E. (1991). *Continuous System Modeling*, Springer Verlag. New York.

Denn, Morton M. (1986). *Process Modeling*. Longman ScientiÞcal & Technical.

Duff, I.S., A.M. Erismann and J.K. Reid (1986). *Direct methods for sparse matrices*. Oxford Science Publications.

Elmqvist, H., M. Otter and F.E. Cellier (1995). Inline integration: A new mixed symbolic/numeric approach for solving differential. algebraic equation systems. In: *ESM.95, SCS European Simulation Multi-Conference*. Prague, Czech Republic.

Hangos, K.M. and I.T. Cameron (2001). *Process Modelling and Model Analysis*; Academic Press, San Diego.

MacKenzie, S. A., P. J. Gawthrop R. W. Jones and J. W. Ponton (1991). Systematic modelling of chemical processes. *IFAC symposium on Advanced Control of Chemical Processes*, ADCHEM'91.

Marquardt, W. (1991). Dynamic process simulation - recent progress and future challenges. *CPC IV*.

Marquardt, W. (1992). An object-oriented representation of structured process models. *ESCAPE-1,* S329-S336.

Marquardt, W. (1994). *Towards a process modelling methodology*. Technical report. Aachen University of Technology. Aachen, Germany.

Moe, Hûavard I. (1995). *Dynamic process simulation: studies on modeling and index reduction*. PhD thesis. Norwegian Institute of Technology, Trondheim.

Ogunnaike, Babatunde A., W. Harmon Ray (1994). *Process Dynamics, Modeling, and Control*. Oxford Uni Press.

Perkins J D, Sargent W H, Vazquez-Roman J and Cho J H (1996). *Computer Generation of Process Models.* Comp Chem Eng, 20 (6-7): 635-639.

Preisig, H.A., (1991a). On Computer-Aided Modelling For Design. *AIChe Annual Meeting*. Los Angeles. 138e.

Preisig, H.A., (1991b). View On The Architecture Of A Computer-Aided Process Engineering Environment. *AIChE Annual Meeting*. Los Angeles. 137b.

Preisig, H.A., Guo, D.-Z. and Mehrabani, A.Z., (1991). Computer-Aided Modelling: A New High-Level Interface to Process Engineering Software. *CHEMECA-91*. New Castle, Australia. 954-960.

Preisig, Heinz A. (1994). Computer aided modeling: species topology. In: Preprints IFAC Symp. *ADCHEM'94*. Kyoto, Japan. pp. 143.148.

Stephanopoulos, George (1984). *Chemical Process Control. An Introduction to Theory and Practice*. Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632.

Westerweele, M.R., H.A. Preisig (2001). Minimal representation of first principle process models. *ESCAPE*. Kolding, Denmark. pp. S47-S52.

Westerweele, M.R., H.A. Preisig M. Weiss (1999). Concept and design of Modeller, a computer-aided modelling tool. *Comp & Chem Eng*, S751-S754.

Westerweele, M.R. (2002), *Five Steps for Building Consistent Dynamic Process Models and their Implementation in the Computer Tool Modeller*, PhD Thesis, TU Eindhoven.