

## Mobatec Modeller - A flexible and transparent tool for building dynamic process models

Mathieu R. Westerweele<sup>a</sup>, Jan Laurens<sup>a</sup>

<sup>a</sup>Mobatec, Veestraat 10, 6134 VJ Sittard, The Netherlands, [www.mobatec.nl](http://www.mobatec.nl)

### Abstract

The software tool that is explored in this paper, **Mobatec Modeller**, builds on a modelling methodology that was originally developed by prof.dr.dipl-ing H.A. Preisig and dr.ir. M.R. Westerweele [1]. With the tool, dynamic process models of any size can be build - from a single unit to entire processing plants (resulting in more than 50.000 equations). There is a wide application range for these models in, for example, research activities, on-line predictions and/or control and operator training simulators.

User-friendliness, functionality and good user-assistance are a prerequisite for being able to manage large models. To demonstrate user-friendliness of **Mobatec Modeller**, several distinguishing features are discussed, such as automatic component distribution, automatic balance equation generation, easily extendable (equation, component, reaction) databases, consistent initial value calculations, code generation for several solvers, very elaborate and useful search functionality.

**Keywords:** **Mobatec Modeller**, Dynamic Models, User Friendliness, Transparent Modelling.

### 1. Introduction

Solving process engineering problems without the help of computer-based tools is for almost any problem an unthinkable proposition. Process simulation, process design, controller design, controller testing, data acquisition and model identification, parameter fitting, valve and pump selection, column sizing are just a few examples taken from a very large catalogue of chemical plant related operations that are almost exclusively done with computer-based tools, nowadays.

Although several good software tools are on the market to build dynamic process models, it often seems that an expert is needed to setup or modify these models. It is also difficult to build models that are transparent for users that were not involved with the model development.

**Mobatec Modeller** [2] is a computer-aided modelling tool to interactively define and modify transparent process models. It can be seen as a shell around existing dynamic modelling packages. It aims to effectively assist in the development of process models and facilitate hierarchical modelling of process plants through a user friendly interface.

With **Mobatec Modeller** process models are constructed from primitive building blocks, being simple thermodynamic systems and connections. It does not, in distinction to existing flow sheeting packages, build on unit models. The “Unit models” that are available in libraries are also built from the primitive building blocks.

**Mobatec Modeller** generates (and solves) symbolic models in the form of differential-algebraic equations consisting of component mass and energy balances, augmented with transfer laws, physical and geometrical property relations and kinetic laws.

## 2. Modelling Methodology

The modelling methodology implemented in **Mobatec Modeller** is based on the hierarchical decomposition of processes (in which material and energy exchange are playing a predominant role during normal operation) into networks of elementary systems and physical connections. Elementary systems are regarded as thermodynamic simple systems and represent (lumped) capacities able to store extensive quantities (such as component mass, energy and momentum). The connections have no capacity and represent the transfer of extensive quantities between these systems. The construction of a process model with this methodology consists of the following steps:

- 1) Break the process down into elementary systems that exchange extensive quantities through physical connections. The resulting network represents the *physical topology*. The process of breaking the plant down to basic systems and connections determines largely the level of detail included in the model. It is consequently also one of the main factors for determining the accuracy of the description the model provides.
- 2) Describe the distribution of all involved chemical and/or biological species as well as all reactions in the various parts of the process. This represents the *species topology*, which is superimposed on the physical topology and defines which species and what reactions are present in each part of the physical topology.  
The species distribution is fully automated by **Mobatec Modeller** and is initiated by introducing species at the battery limits of the modelled process.
- 3) For each elementary system and each fundamental extensive quantity (component mass and energy) that characterises the system write the corresponding balance equation.  
**Mobatec Modeller** automatically generates all the needed balance equations for component mass and enthalpy of each system, since these balances can be trivially formed from the model designers' definition of the physical and species topology of the process. The user cannot edit the generated balance equations!
- 4) Add algebraic equations to the model definition:
  - Choose the transfer laws and kinetic laws that express the flow and production rates that appear in the balance equations.
  - Express the fundamental extensive variables that characterise each system as a function of intensive variables characterizing the same system.
  - Look for dependencies between the intensive and geometric variables that have been introduced and write these dependencies out as equations of state.

The dynamic balance equations (step 3) and the algebraic equations, which are placed on top of the physical topology and species topology, represent the *equation topology*.

- 5) Add the (dynamic) behaviour of the information processing units, such as transmitters, adjusters and controllers.

These steps for building a model do not have to be done strictly in this sequence - at least not for the overall model. It is left to the model designer when the details are being specified in each part of the model.

This five step procedure of building dynamic process models always results in a set of differential algebraic equations (DAE) with an index of one (i.e. structurally solvable by most available solvers). The model can be used for solving certain problems related to the process or it can be further modified by applying mathematical manipulations, such as linearization or model reduction.

### 3. What is User-Friendliness?

According to Wikipedia [3] “usability (i.e. user-friendliness) is a term used to denote the ease with which people can employ a particular tool or other human-made object in order to achieve a particular goal. In human-computer interaction and computer science, usability usually refers to the elegance and clarity with which the interaction with a computer program or a web site is designed.”

Imported considerations for user-friendliness include: Who are the users? What is their general background? What do they want or need? What is the context in which the users are working? Can users accomplish intended tasks at their intended speed? Is working with the tool easy to learn? Etc.

When people speak about a tool that is “easy to use” or “user-friendly” they often actually mean that it is “easy to learn” or “easy to use without learning”. Easy to learn means that if you sit somebody with no experience down with a tool, it will be (relatively) easy for them to learn how to use it.

Useful aspects for learnability are ‘intuitiveness’ and ‘obviousness’. If something is intuitive or even obvious, it’s far easier to learn than if it is not. Unfortunately intuitiveness and obviousness are very subjective and often depend on background knowledge; different people find different things intuitive or obvious. Luckily, the group of users we are aiming at in this case roughly (should) have the same background and interests.

Another important aspect of user-friendliness of a tool is its documentation. Good documentation is indispensable. It should be correct, up-to-date and cover a reasonable range of the functionality. Unfortunately, typical users will not read the documentation until they are “in trouble”. Therefore the documentation should be kept as short and as to the point as possible. A very good way of “documenting” the functionality of a tool is to make use of ‘Tool Tips’ that appear, for example, after your mouse cursor hovers for a few seconds or when the right mouse button is pressed on an item in a toolbox.

#### 4. User-friendly Interface of Mobatec Modeller

This section discusses some aspects of the implementation and user interface of **Mobatec Modeller**, a computer-aided modelling tool with the objective to provide a systematic model design method.

A software tool cannot reach to its full potential if the software developers of the tool do not exactly know what problems the end users want to solve and what questions these users want to see answered. More often than not the end users of a tool and the developers of this same tool live in two different worlds (computing, implementing vs. engineering, chemistry, academics) and have a hard time communicating with each other. This, of course, makes the software development difficult, slow and time consuming.

We found ourselves in the fortunate situation that we were (and still are) doing industrial projects (for several companies) whilst, at the same time, being able to develop the software that is used for the implementation of these projects. The best way of finding out what the user wants, is to be a (real!) user yourself.

During the PhD study of M. Westerweele the focus was on the algorithms behind the mathematical modelling of physical/chemical processes. Afterwards, the focus needed to be shifted to applicability, needs of industry and user-friendliness.

The focus of **Mobatec Modeller** is primarily on modelling and not on problem solving (although both activities are supported in the tool). Most of the currently available modelling languages and simulation packages focus on the manipulation, specification, analysis and/or solution of existing or predefined models and more or less leave out the model building part. In general it is assumed that the mathematical model of the process under investigation is known or easy to assemble. The development of process models, however, is slow, error prone and consequently a costly operation in terms of time and money.

Dynamic modelling is an acquired skill, and the average user finds it a difficult task. A modeller may inadvertently incorporate modelling errors during the mathematical formulation of a physical phenomenon. Formulation errors, algebraic manipulation errors, writing and typographical errors are very common when a model is being implemented in a computing environment. **Mobatec Modeller** automates several of the needed modelling operations and therewith eliminates a lot of simple, low-level (and hard to detect) errors.

The modelling methodology implemented in **Mobatec Modeller** also forces the model builder to make assumptions that are closer to the “physical reality” than when more conventional tools are used. Take, for example, the very well known and often used model of a CSTR with a flow in and out. The commonly used assumption that the liquid volume is fixed holds several other assumptions (e.g. fixed density and infinitely fast adaptation of the outflow to changes in the inflow) of which the implications are often not clear to novice modellers. While this approach is nice for academic examples, since the resulting set of equations can be solved algebraically, it is hardly ever appropriate when building dynamic models for ‘real’ processes. It also tends to let novice modellers make wrong sequential assumptions when the model becomes a bit more elaborate. **Mobatec Modeller** therefore promotes the use of flow equations that depend on the states (e.g. the pressure or the level) of the interconnected systems.

We found that “typical” model engineers are not particularly interested in the exact mathematics behind the solving of a problem. They do, however, often want to know how a model or sub model is setup (Which equations are used? What are the assumptions?). They want to be able to understand the used model and possibly make certain adaptations to make it better fit their needs. **Mobatec Modeller** therefore supports an open model structure in which all used equations can always be viewed and edited (except for the balance equations, that are generated automatically and purposely cannot be edited). A drawback of several (dynamic and/or steady-state) flow sheeting and modelling packages is that the equations of predefined sub models (i.e. units) cannot be edited. In some cases the accompanying documentation lists (partially) which equations are implemented in the sub model, but that is as far as it goes. The disadvantage of this approach is that it is not flexible and a lot of documentation is needed. To make a sub model more flexible, either a lot of options need to be implemented (implying a lot of overhead to be carried along with the model) or a lot of slightly different sub models need to be available. Whilst it is, of course, handy and even necessary to have a large database of predefined sub models (i.e. unit operations); it is not user-friendly to confine the user to the particular implementation of the model builder.

To promote user friendliness, **Mobatec Modeller** has several distinguishing features:

- **Context sensitive, non-modal properties toolbox.** A rather annoying feature of many programs is the use of so-called modal dialog boxes for displaying properties. This means that the user must first close the box before he can continue with the program. This can be very inconvenient if the user wants to view the properties of several objects. In **Mobatec Modeller** the properties toolbox directly adapts to the active selection.
- **Unlimited Undo/Redo functionality.** All actions can be undone and redone, if necessary. This eliminates the need for modal dialog boxes with questions like “Are you sure you want to ...”
- **Visible time-scale assumptions.** On a high level, the screen of **Mobatec Modeller** may resemble a typical flowsheet. It is, however, always possible to zoom in to a lower level in which the user can graphically see which time-scale assumptions have been made in each sub model. Due to the used modelling methodology, all mass transfers, heat transfers, phase transitions, etc. must be explicitly visible. A (sub) model is always an interconnected network of systems.
- **Automatic species distribution.** The definition of the species topology is initialised by assigning sets of species and reactions to some elementary systems. To aid in this definition, species and reaction databases are used. The user may also specify the directionality (i.e. uni or bi-directional) and permeability (i.e. selective transfer of species) of individual mass transfer connections. The distribution of the species over all systems is automated and uses the facts that assigned species can propagate through permeable mass connections and species may generate “new” species (via reactions), which in turn may propagate and initiate further reactions.
- **Ample user preference settings.** User-friendly also means that a user can do his work “his own way”. The working environment can therefore be adjusted via several user preferences and there are several ways to perform actions: via the toolbar, menus, context-sensitive popup menus or single-key keyboard shortcuts.
- **Powerful search functionality.** **Mobatec Modeller** has an elaborate and useful search functionality, which is especially handy for larger and more complex models.

The user can easily find improper equation sets, uninitialized variables, duplicate names, species, reactions, equations, variables, domains, properties, etc.

- **Easily extendable databases for equations, variables, species and reactions.** All used equations only have to be defined once. Equations are parsed as they are defined. Low-level errors are therefore minimized and the variables that are used in the equation are automatically detected.
- **Automatic equation sorting and, consequentially, declaration of constants and variables.** Due to the division of the model into a set of interconnected systems, typically, only somewhere between one and ten equations are associated with each modelling object (system, connection or reaction). The equations can be selected from user-extendable databases. The user is helped with defining a set with an equal number of equations and unknowns. **Mobatec Modeller** automatically determines which variable should be solved from which equation and places the used constants and variables of each object in the correct declaration lists.
- **Consistent initial value calculations.** For dynamic simulations a consistent starting condition, the so-called initial values for all defined variables, needs to be present in order for the solver to find a solution. This, often cumbersome task, is easy with **Mobatec Modeller**. The variables of an object are related to each other via the defined algebraic equations. Therefore their values can, in principle, not be chosen independently. The initial values of each object can be calculated from a set of independent variables.
- **Code generation for several solvers.** Apart from the code that can be generated for the integrated equation solver, **Mobatec Modeller** can make input for several other problem solver tools, such as Aspen Custom Modeler, gProms, Matlab, e-Modeler, Modelica, or any other DAE solver.
- **Frequent updates.** Our software is constantly further developed and about twice a month a new release is available. Since a compact programming style is used, the size of the installer is small and the updates take less than a minute.

## 5. Conclusion

**Mobatec Modeller** is a user-friendly tool to define and modify dynamic process models. Even beginning users can relatively quickly setup rather complex models that are transparent for others without much documentation. The constructed models can be used in several existing problem solver tools.

## References

- [1] Westerweele, M. R.: 2003, "Five Steps for Building Consistent Dynamic Process Models and Their Implementation in the Computer Tool Modeller", PhD thesis, TU Eindhoven, Eindhoven, The Netherlands.
- [2] Information on Mobatec Modeller and technical consultation: [www.mobatec.nl](http://www.mobatec.nl).
- [3] Wikipedia. Multilingual, web-based, freecontent encyclopedia project. [www.wikipedia.org](http://www.wikipedia.org).