Model-based training and analysis with the CiT-Observer

Dr. Michael Wulkow, Regina Telgmann

Overview

- Motivation
- Principle of data exchange
- Starting point
- Building up an observer model
  - Administrator mode
  - Controls
  - User mode
- Extensions

The CiT-Observer can be applied to

- Predici
- Presto-Kinetics
- Parsival

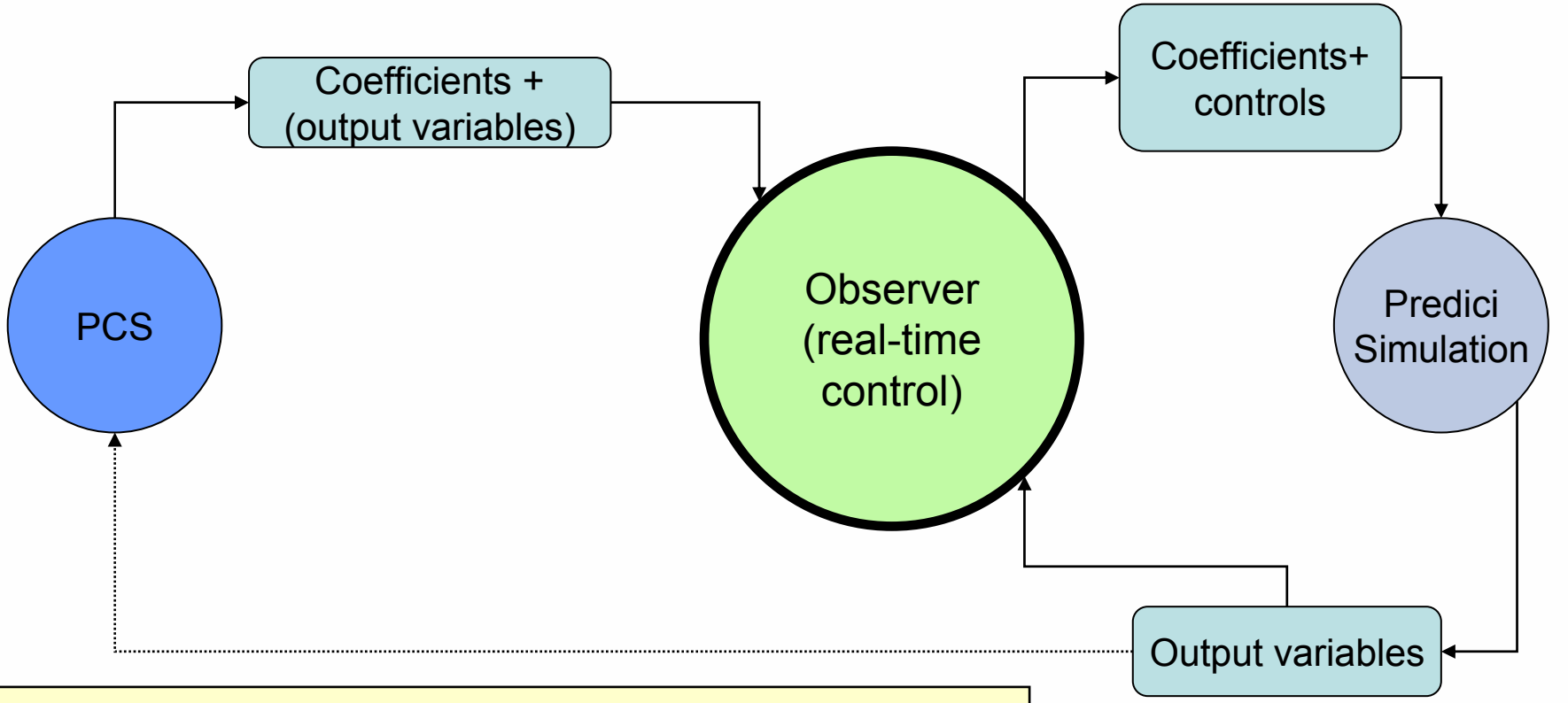
The extension which is necessary to connect a CiT modeling tool to the CiT-Observer is called **simulation server**. The simulation server restricts the time step control by enforcing a simulation speed related to real time and at the same time communicates with the observer by exchange of input and output files. Thus the simulation server can also be used in connection with third party programs.

Motivation

- Training and analysis of
  - product changes
  - transient behavior of a reactor or plant
- Comparison of model and process control system (PCS)
- Online observation
- Model predictive control, process intensification
- Leaving a model with an easy-to-use and save interface to people not used to the modeling tool (“customized Predici view”)

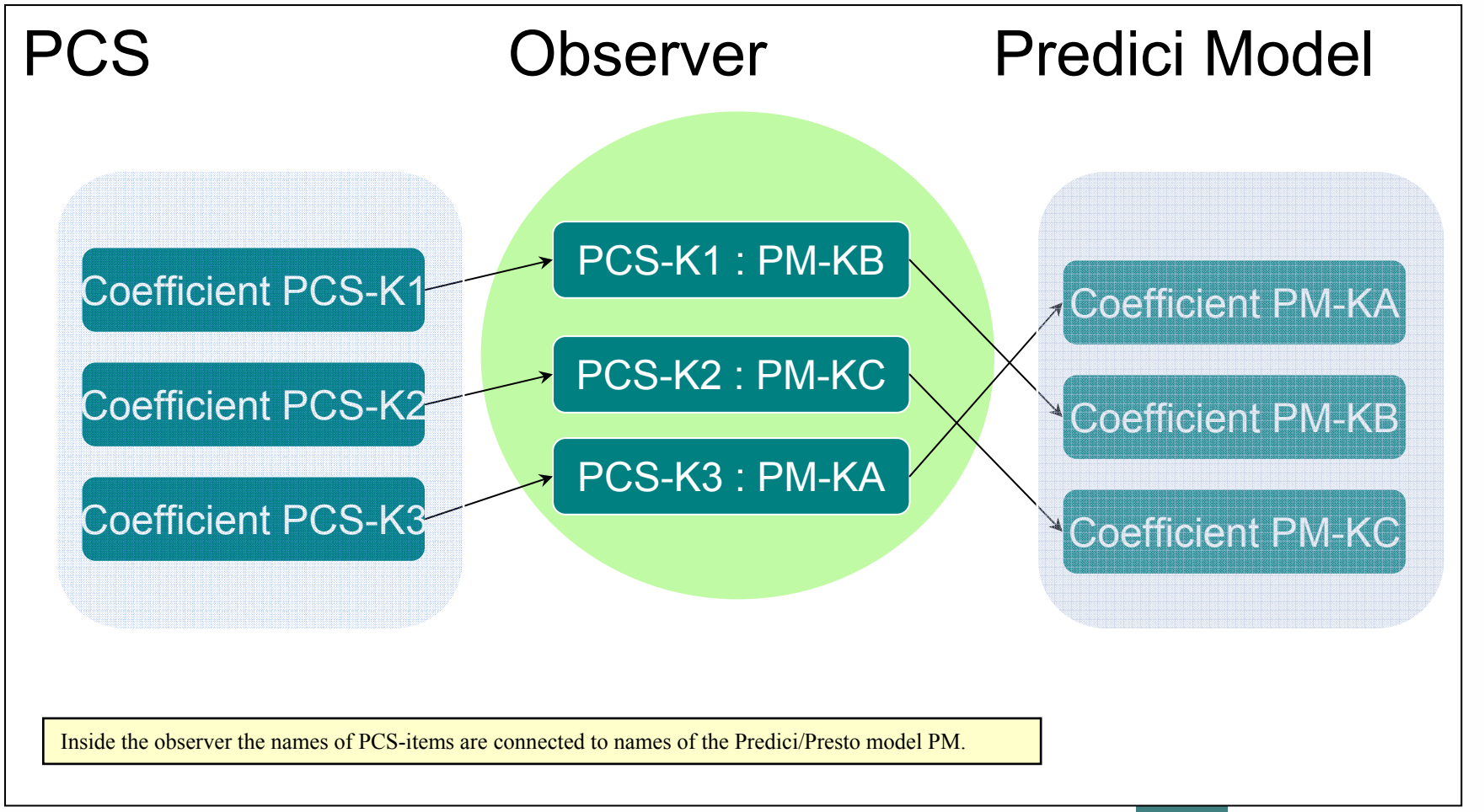
Up to now, two concepts have been applied in order to leave models developed in CiT-tools to users not involved in the modeling. The ModelTrainer provided a fixed implemented interface of a model, whereas the OLE/Com-interface left the control of Predici to an in-house programmer (e.g. using VB). In both cases Predici performs simulations as quick as possible and does not allow intermediate input without stopping of the simulation. The CiT-Observer extends both concepts in view of flexibility and user-friendliness.

Data exchange

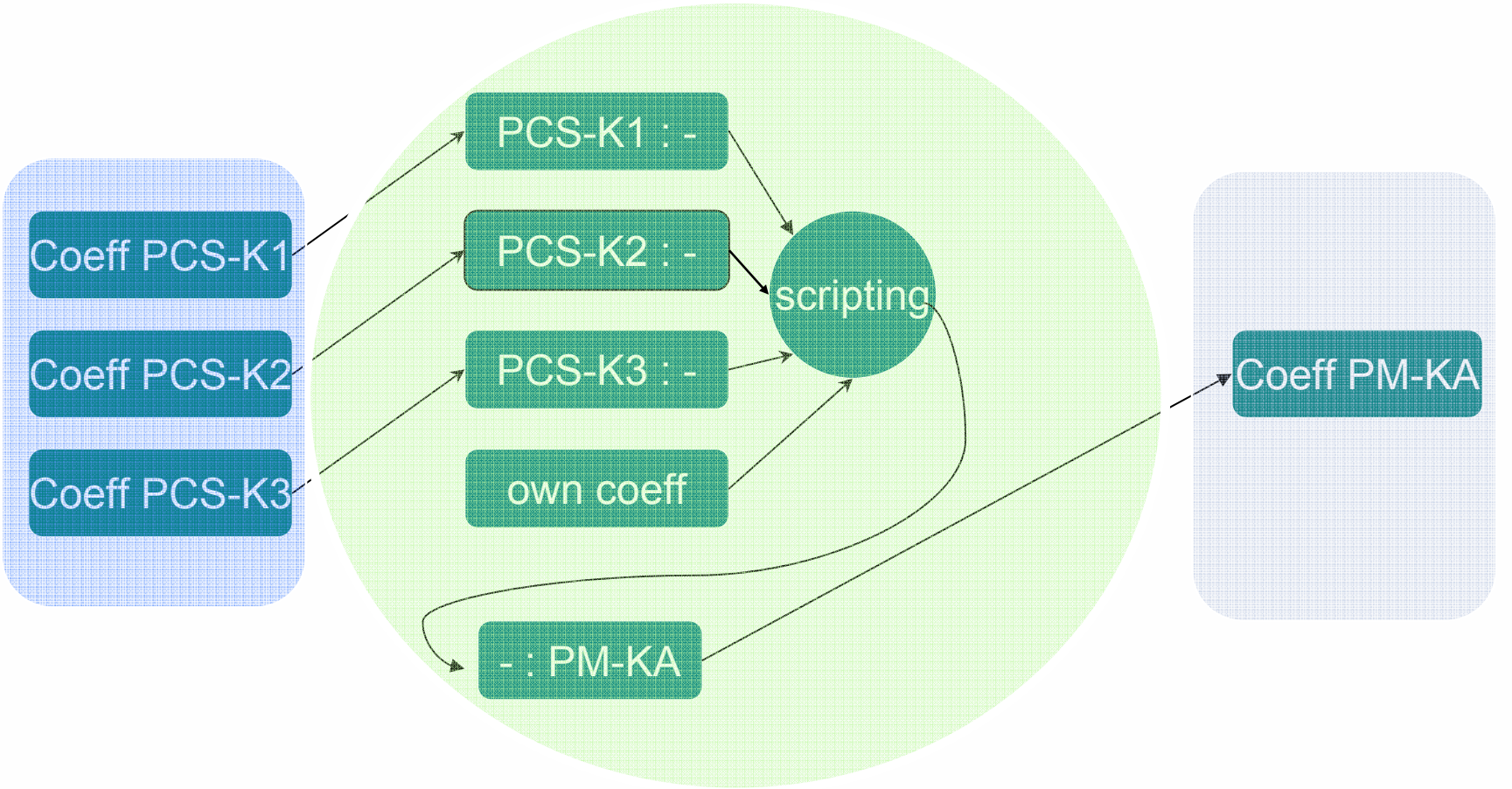


The process control system (PCS) provides a text file with coefficient and output values, including PCS names. The observer reads, presents and passes the coefficients through to the simulator's input file. Predici/Presto provides output variables, including internal model names, which are also presented by the observer.

Data flow of coefficients, direct 1:1

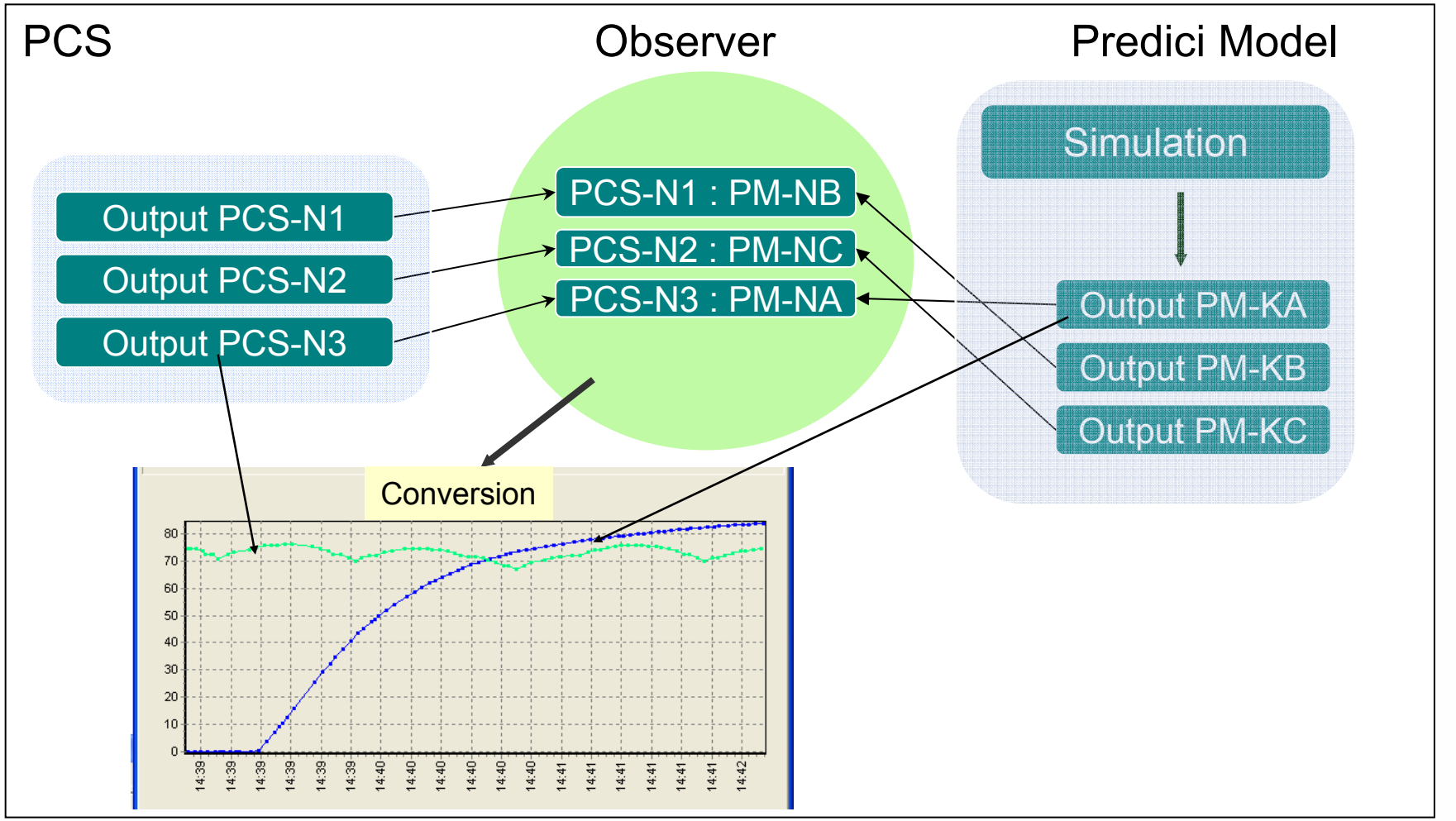


Data flow of coefficients, indirect n:m



The interpreter allows the re-calculation (by means of arithmetical operations) of PCS-items as well as Predici/Presto items, e.g. unit conversions, summation of several items, etc. All other calculation have to be performed inside the Predici model.

Direct comparison: output variables



Inside the observer the names of PCS-items are connected to names of the Predici/Presto model PM. So a direct comparison of output from the PCS and simulation output is possible! This evaluates your model quality!

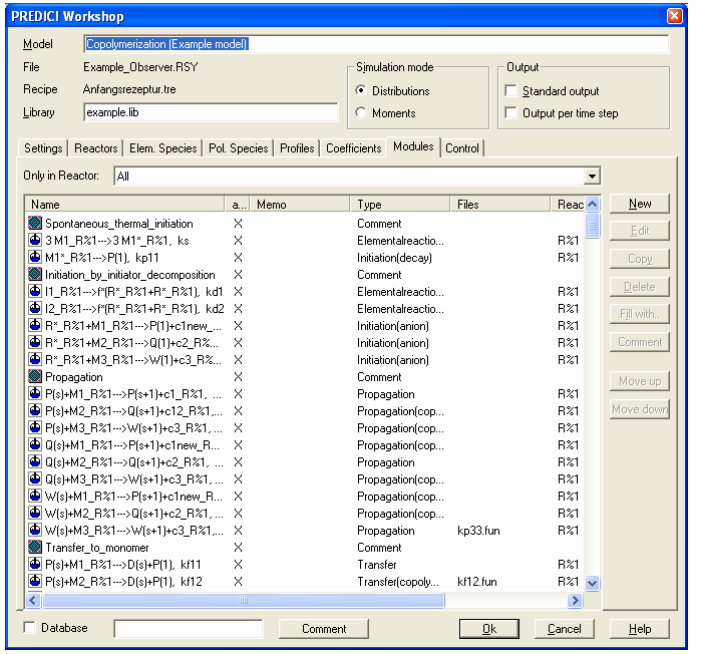
Starting point for observer project

- Validated kinetic or process model
- Known range of application
- Sufficiently predictive
- Implementation of model in CiT-tool with
  - User-defined output functions (library)
  - Control of simulation in terms of parameters
  - Real-time simulation speed possible

When a model is prepared for use in connection with the CiT-Observer, it has to be considered, that the observer only controls parameters/coefficients of a model and reads the values of output functions. This does not lead to a functional restriction (nearly anything can be controlled by parameters), but makes the communication simple and transparent.

Building up an observer model – model preparation

- Predici terpolymerisation model in three reactors



Vol_R%1			1.0000e+04
Vol_R%2			1.0000e+04
Vol_R%3			1.0000e+04
Feed_Monom...	in_kg/s		1.0000e+02
Feed_Monom...	in_kg/s		1.0000e+02
Feed_Wasser	in_kg/s		0.0000e+00
Feed_Initiator	in_kg/s		5.5000e+00
Feed_Monom...	in_kg/s		0.0000e+00
Feed_Monom...	in_kg/s		0.0000e+00
Feed_INI1_R%3	in_kg/s		0.0000e+00

(excerpt from list of coefficients)

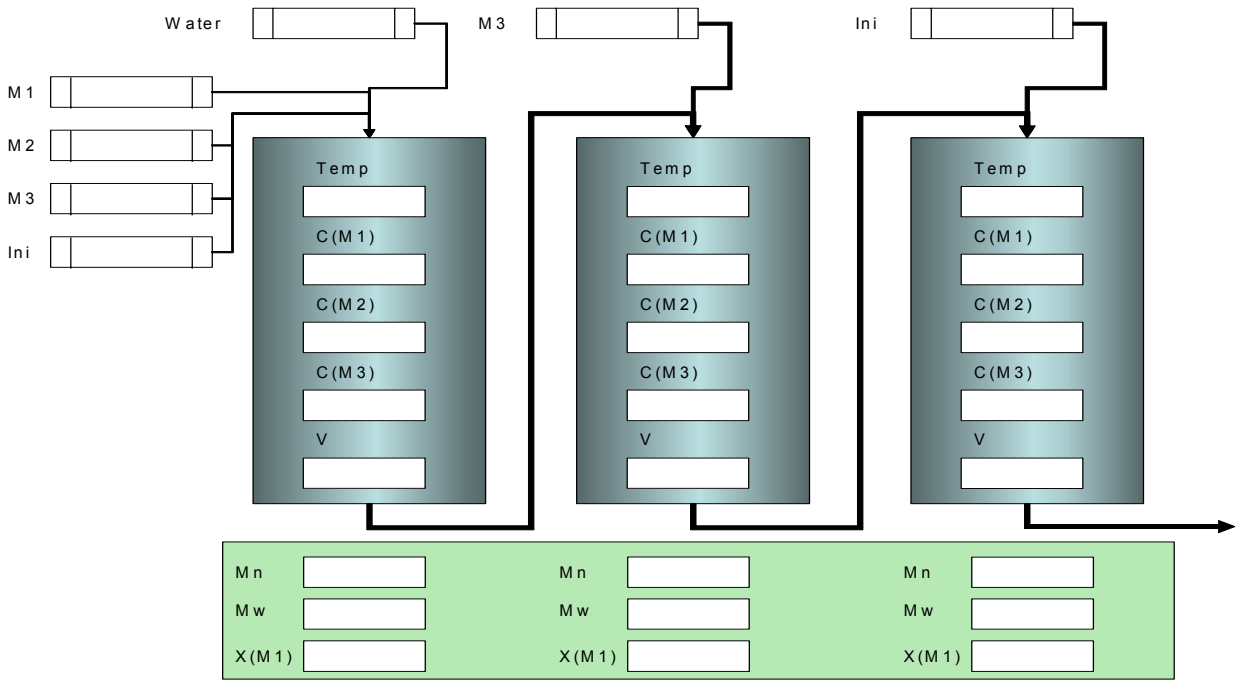
Temp_R%2	x	Temp_R%2.fun	Output (standard)
Temp_R%3	x	Temp_R%3.fun	Output (standard)
C_M1_R%1	x	C_M1_R%1.fun	Output (standard)
C_M2_R%1	x	C_M2_R%1.fun	Output (standard)
C_M3_R%1	x	C_M3_R%1.fun	Output (standard)
C_M1_R%2	x	C_M1_R%2.fun	Output (standard)
C_M2_R%2	x	C_M2_R%2.fun	Output (standard)
C_M3_R%2	x	C_M3_R%2.fun	Output (standard)
C_M1_R%3	x	C_M1_R%3.fun	Output (standard)
C_M2_R%3	x	C_M2_R%3.fun	Output (standard)

(excerpt from library)

The model is developed in Predici in the usual way. A series of coefficients is defined in order to control volumes and feed streams. In the library, output functions for all interesting values, indices and state variables are provided. Predici writes all output variables into an output file, ready to be read by the observer

Building up an observer model - graphic

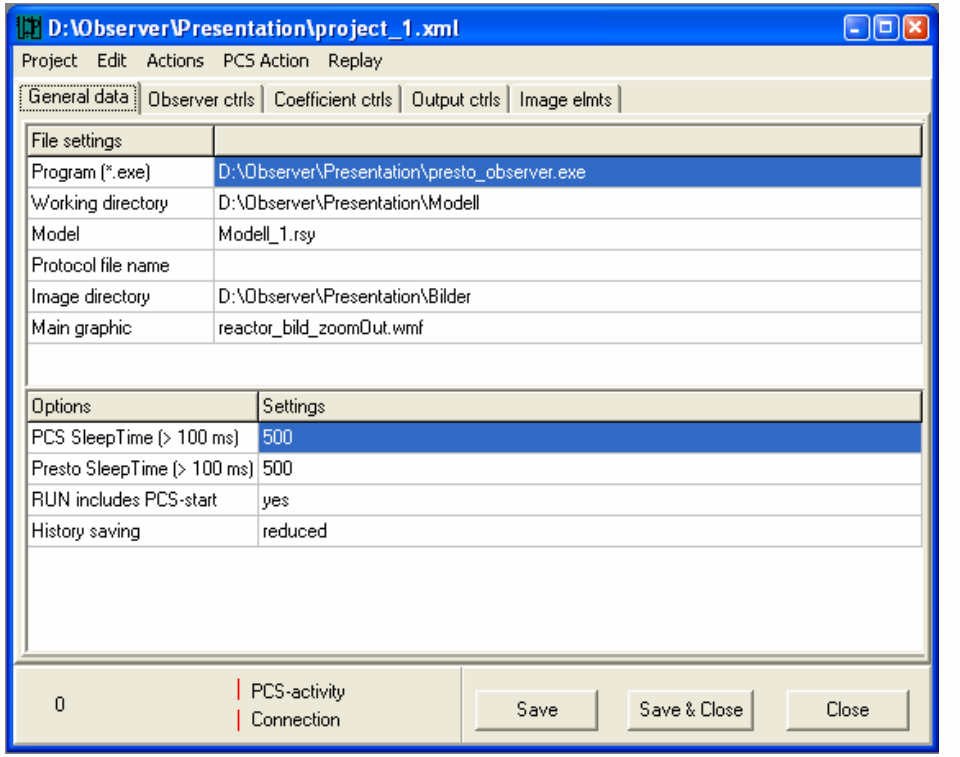
- Schematic picture of the plant, drawn in MS PowerPoint™ and stored in wmf-format (scalable)



Graphics can be imported into the CiT-Observer. The meta file format used in PowerPoint has advantages, since it can be scaled easily, whereas bitmaps may lead to a bad quality if resized.

Administrator mode – creating a new observer project

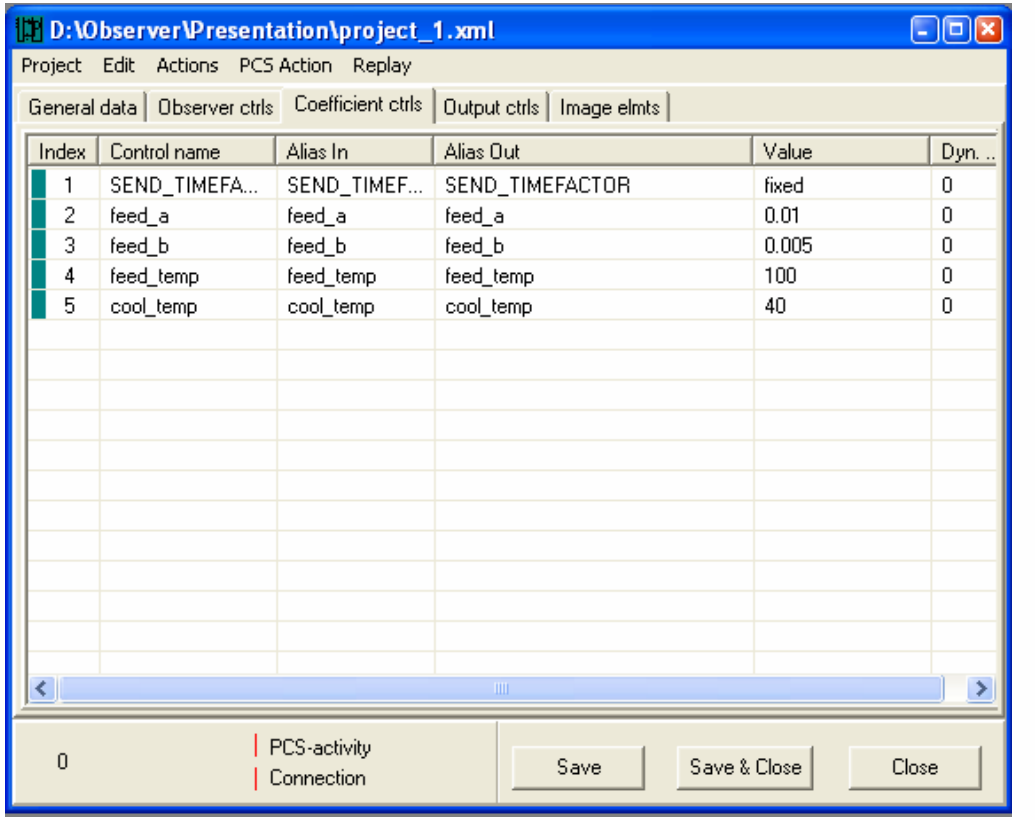
- Observer main control form



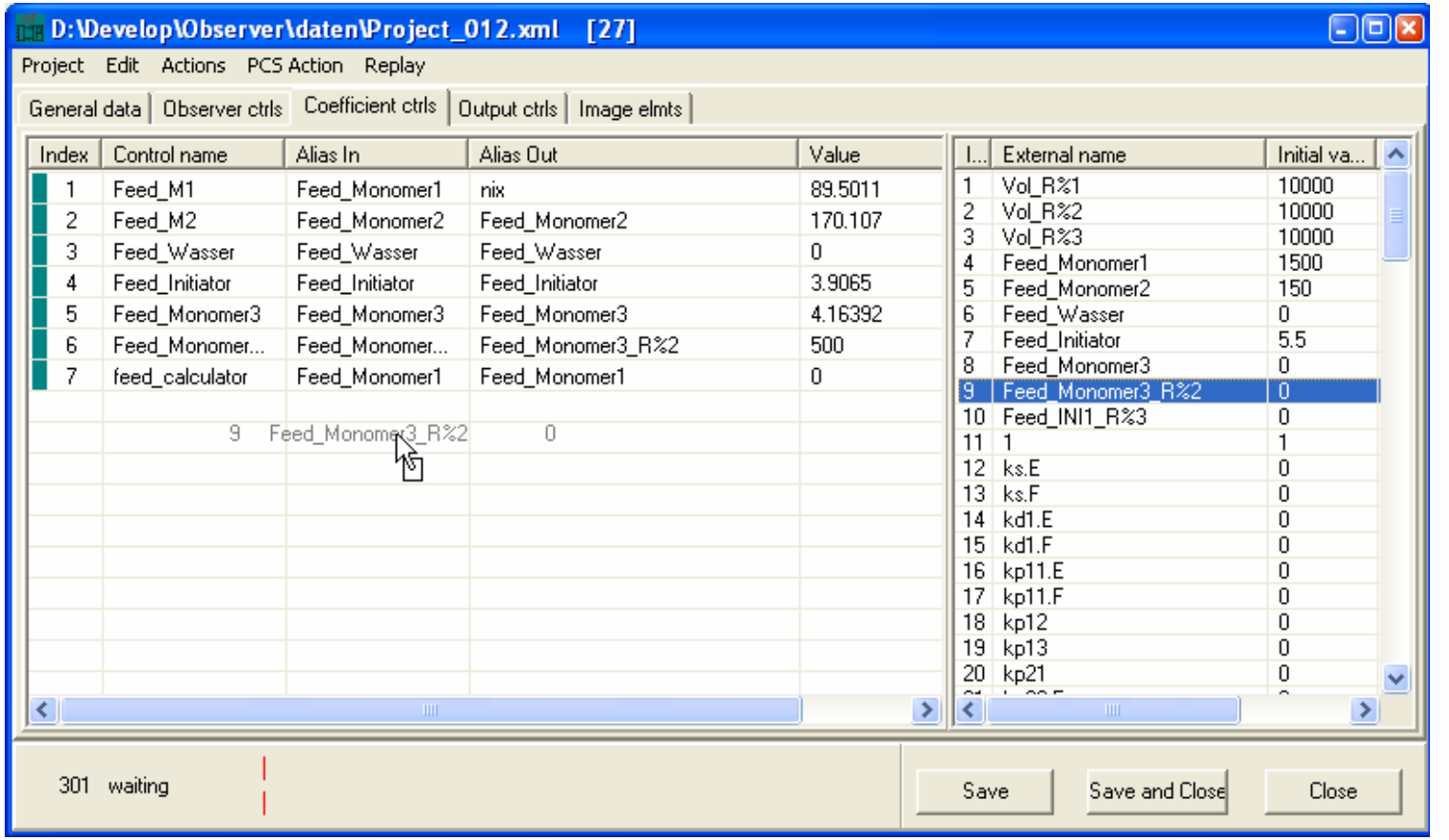
In the above control form, a series of settings has been done: the used tool, the model, the working directory and the background graphics have been set. Any stored observer project can easily be loaded.

Observer items definition

- PCS-coefficient name = alias-in
- PM-coefficient name = alias-out



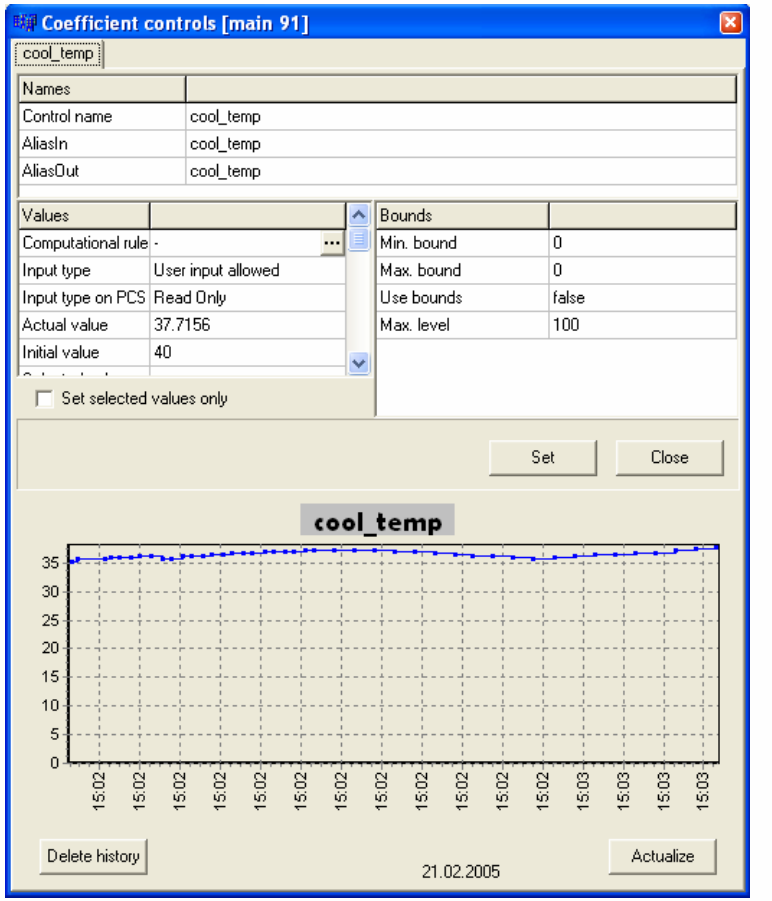
Import of Predici model names



The input of Predici model names and coefficient initial values is supported by drag&drop from the mentioned Predici model.

History of coefficients

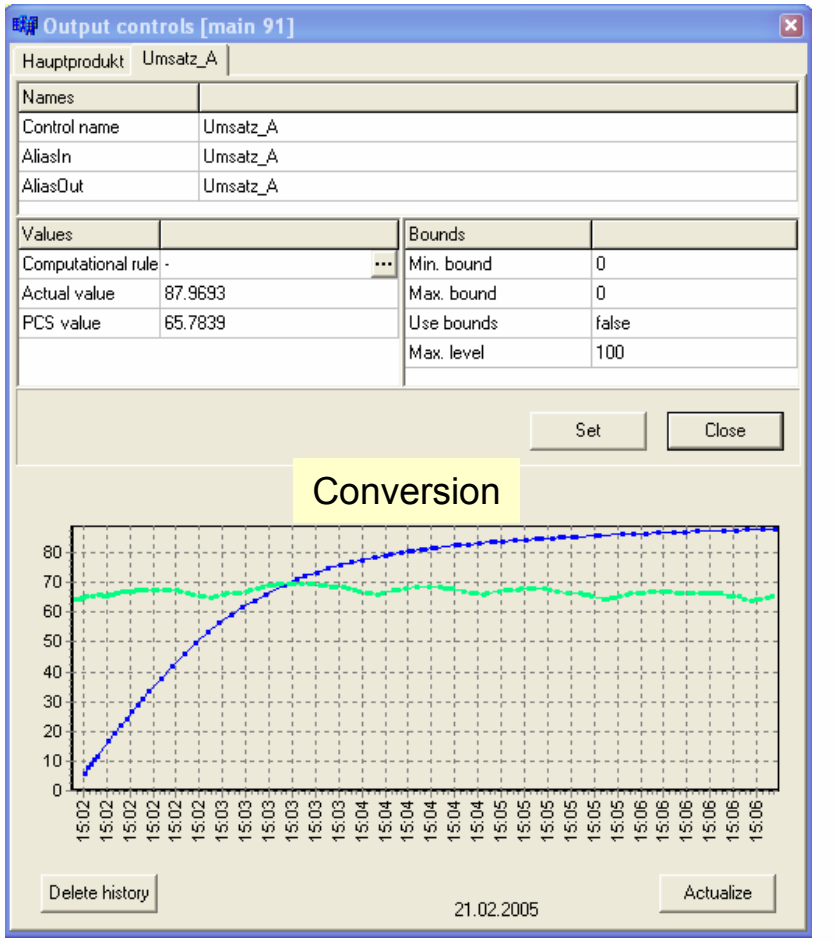
- Observer item dialogue
- Controls properties
- Shows initial and actual values
- Allows user input for actual values
- Presents the history
- Absolute times presented



Each observer object/item controls its properties like boundary values or “Read only”-type. The history during a run can be saved, presented and compared to histories of other scenarios!

History of output variables

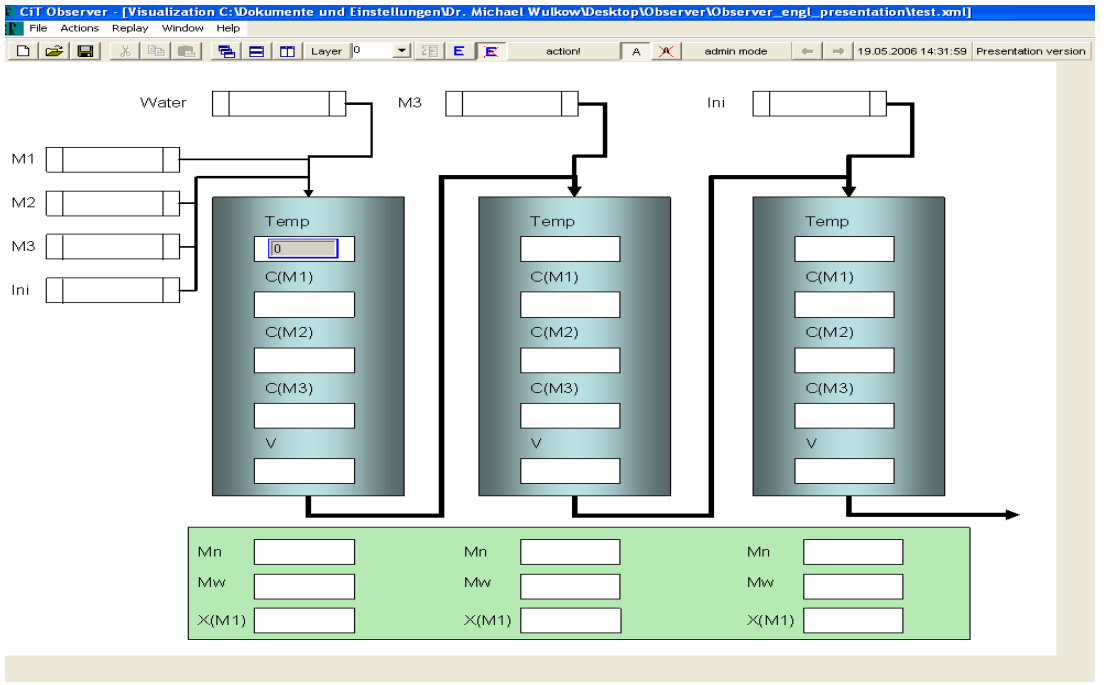
- Observer item dialogue
- Controls properties
- Presents the history
- Absolute times presented



Each observer object/item controls its properties like boundary values or “Read only”-type. The history during a run can be saved, presented and compared to histories of other scenarios!

Loading a background image

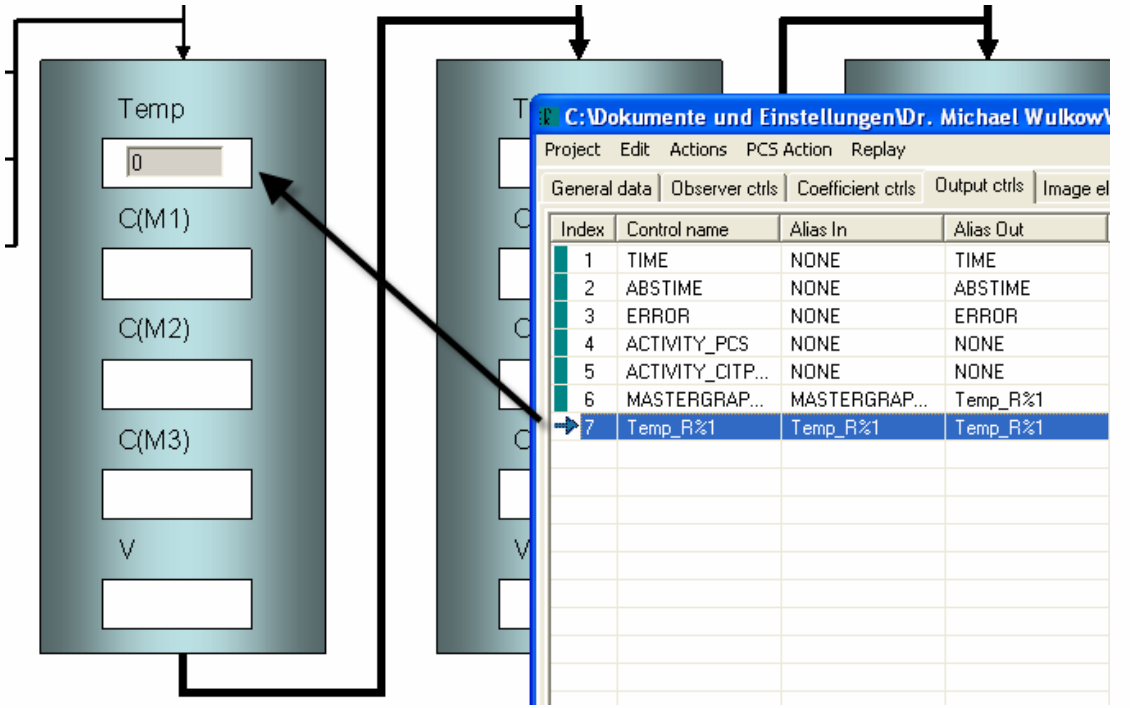
- On menu item, the process graphic appears in the observer environment (in user-mode this happens automatically)



In one observer project an arbitrary number of graphics can be loaded. It is possible to switch from one picture to another by a user-driven event. Thus an online change of process configuration can be visualized.

Adding controls to the image

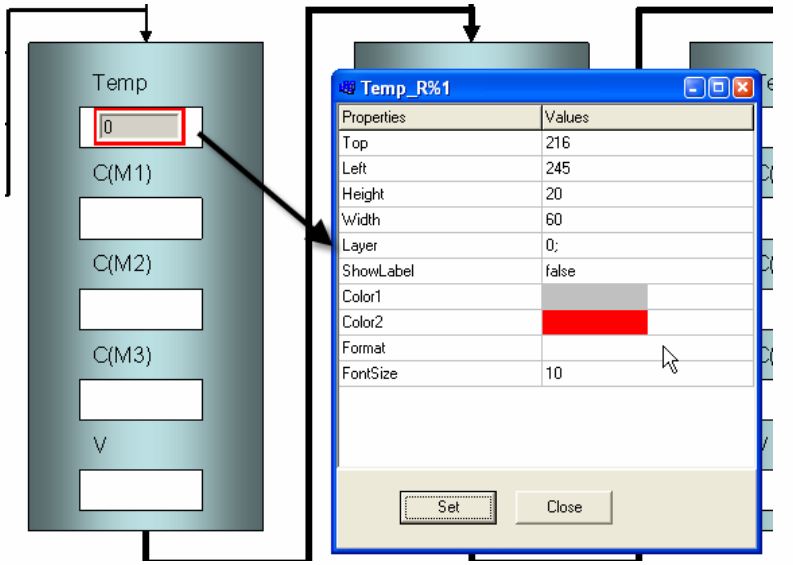
- Drag & drop of an observer item to a reasonable place on the image is the easiest way



The same list is also used for the output of all values during a simulation. The string ”by rule” indicates, that this item is a result of a re-calculation.

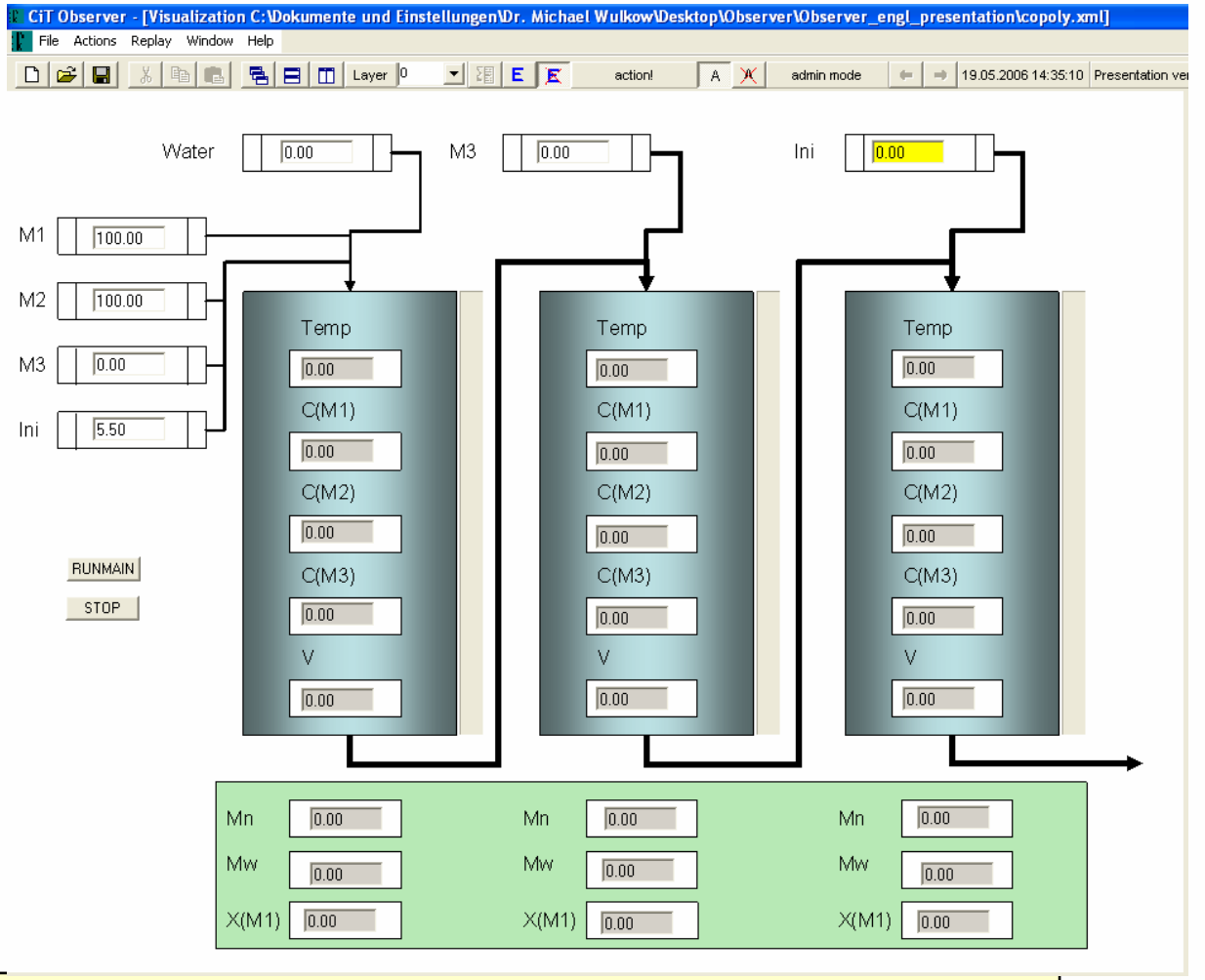
Adjusting controls

- Each control is supported by a so-called “inspector”. The inspector allows to set all instructions concerning the graphical aspects like output formats, colors and sizes.
- Buttons, bars, boxes, graphics available
- Controls can be placed on so-called “layers”



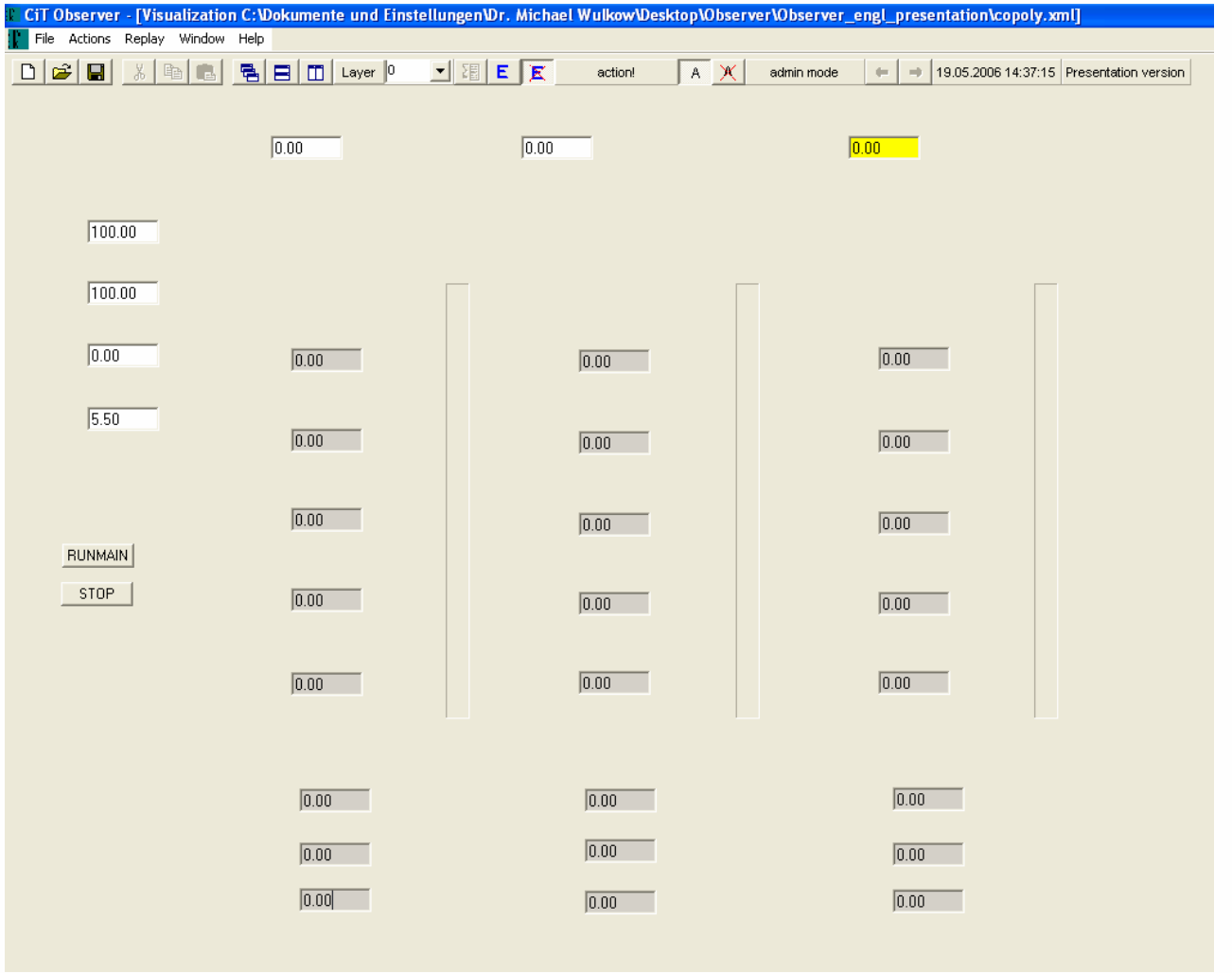
The options listed in the inspector show a wide range of applications. For example, it is possible to change the output color, if a value leaves a prescribed range.

The full control panel…



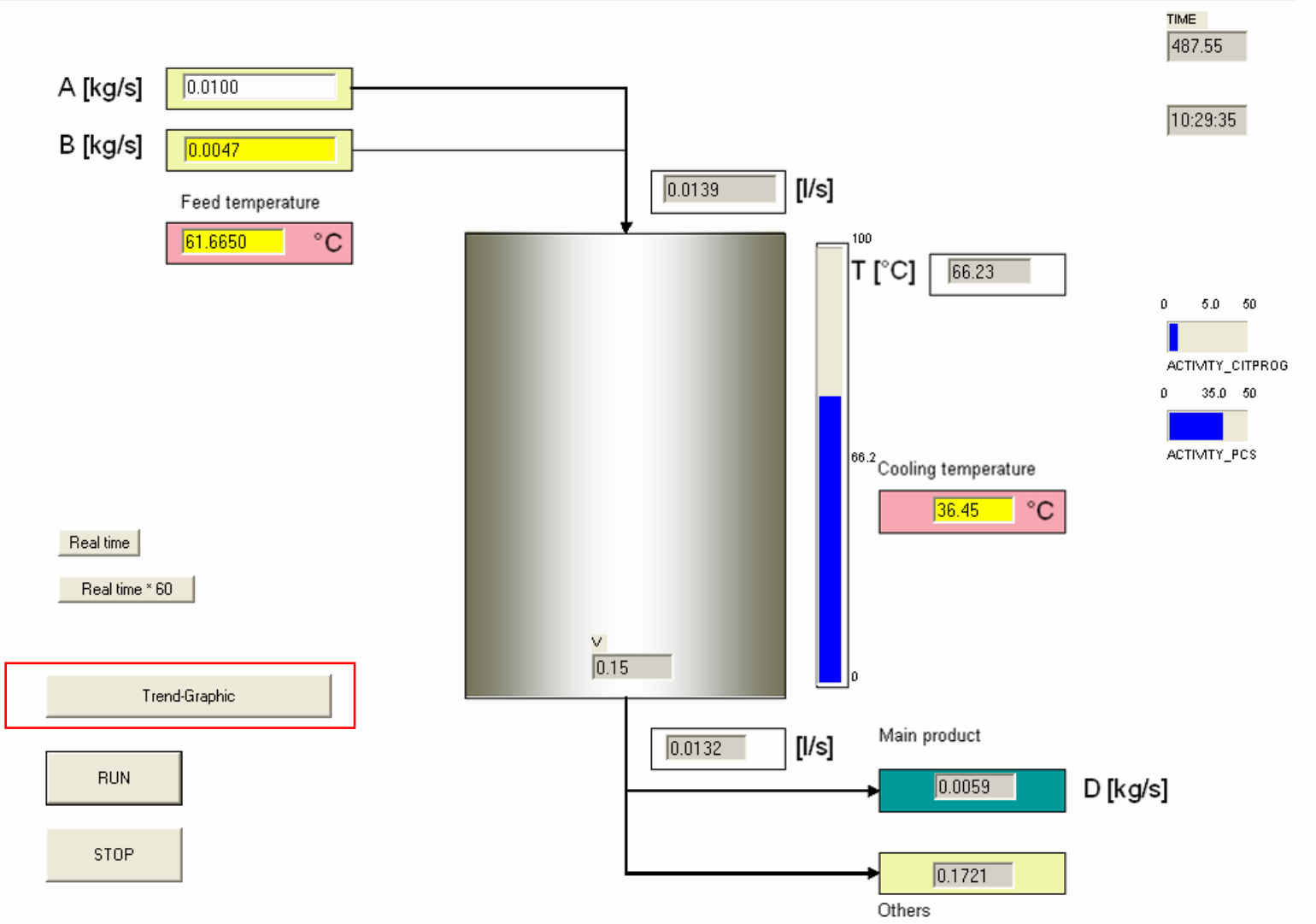
In the full control panel, a series of output fields is used in order to show concentrations, product properties and temperatures. Input fields are used to set flow rates, feed temperatures, resp.. Status bars show the progress of the level control (PID) in the three reactors. Finally a series of buttons and bitmaps activate run time settings.

The full control panel without background



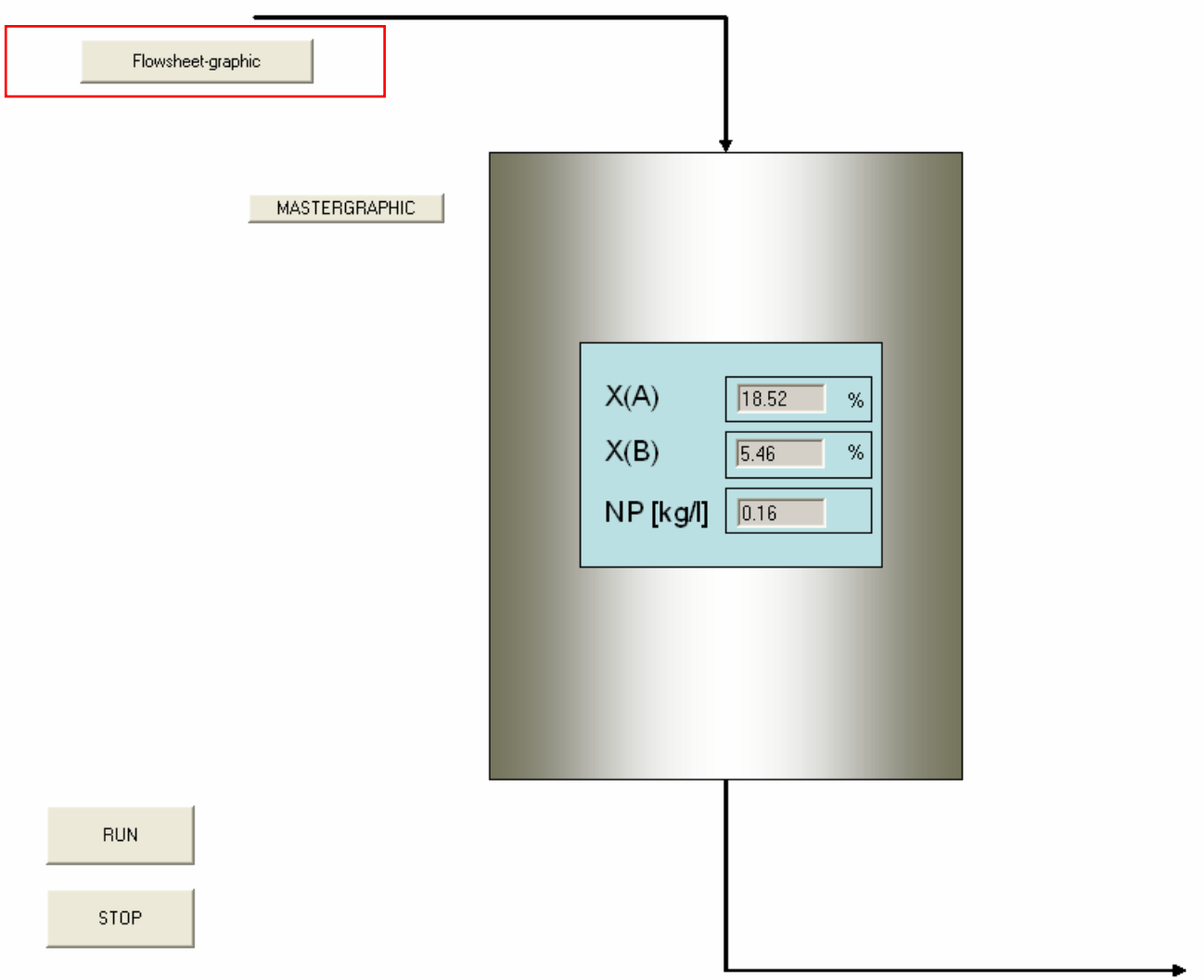
There are several options to hide and show graphical elements. The above picture shows the pure control elements added by the observer administrator.

Another control panel with layers, Layer 1: overview



Each control may change layer and activate new picture…

Layer 2, detailed information



Principles of the process control

- Starting an observer project in the user mode
- All control settings are protected, only allowed input can be made
- Start of the simulation
- Any editing in the observer interface is directly transferred to the simulator and there processed after the next time step
- The simulator runs related to real time and writes all values into a file
- This file is read by the observer permanently, all values are shown in the respective controls

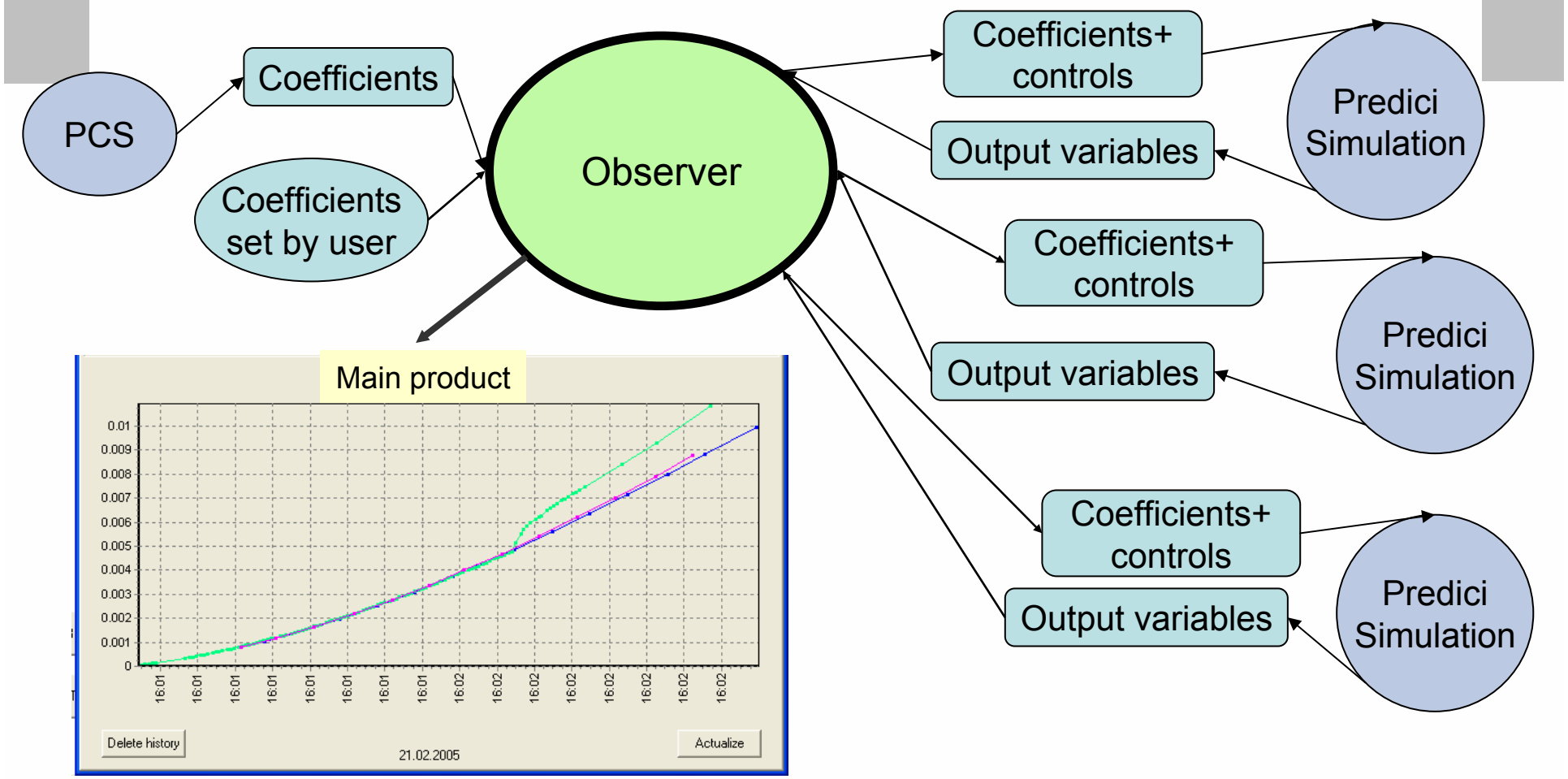
Of course, it is also possible to run an observer session directly in the administrator mode. However, the final application will mainly be done in a special user run-time version of the CiT-Observer.

Different scenarios

- You can start (and compare) an arbitrary number of projects, each observing an own Predici/Presto model and Predici/Presto session, each reading the same PCS data
- A “twin control” or “what-if” starts the same Predici/Presto model with initial values given by the main project at time t_{actual}, without PCS-control → user controlled simulation

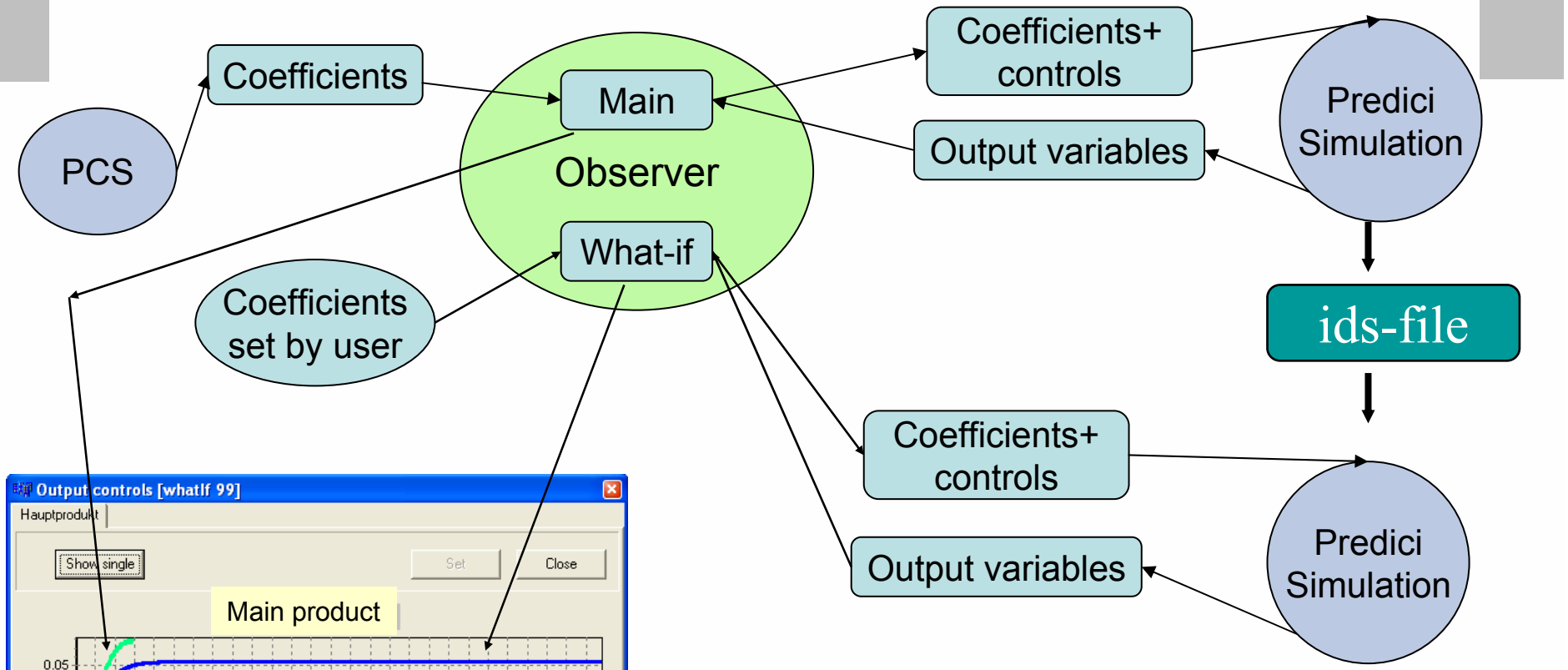
Of course, it is also possible to run an observer session directly in the administrator mode. However, the final application will mainly be done in a special user run-time version of the CiT-Observer.

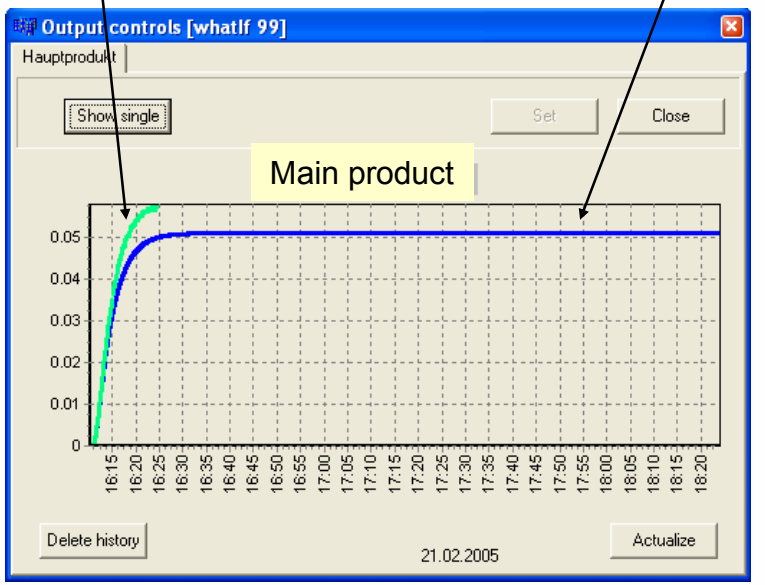
Parallel main scenarios



It is possible to start an arbitrary number of parallel sessions (each one run-time license). The results of each are shown as in the main scenario, but it is also possible to present the results of equally named properties simultaneously (!) in the history dialogue. Therefore direct comparison of different models or different parameter strategies is an easy task.

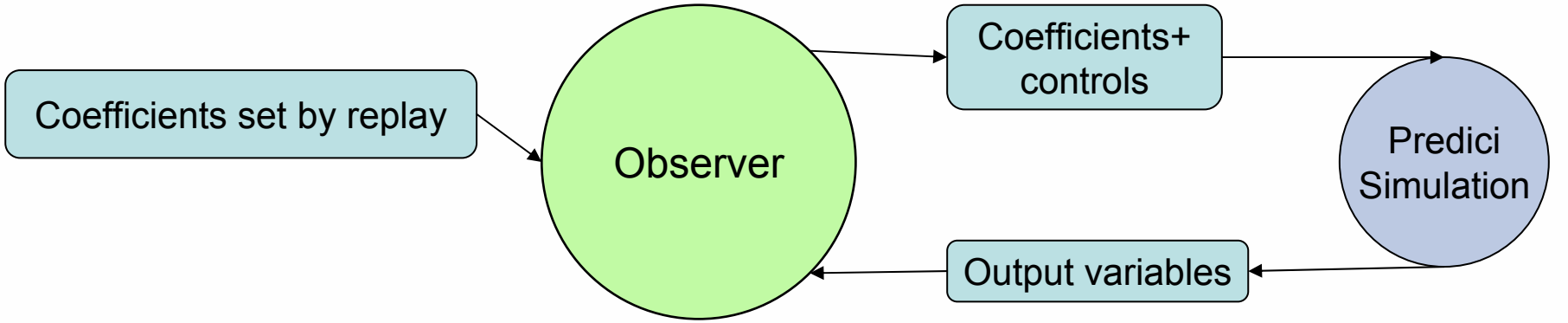
What-if-Scenario





Starting the what-if-scenario from the main scenario at a time t implies the writing of an ids-file with the actual results and starting of the same or a compatible model, reading the ids.file and performing from time t on. In the what-if-scenario the control of the coefficients is performed by the user. If the simulation of the what-if runs faster than real time you have a look what will happen in the future compared to the actual process.

Replay



The replay plays a list of stored instructions, set either in a former main scenario or edited by the user and therefore reproduces former strategies. Editing allows to import disturbances of a strategy which can be used for teaching purposes.