



Expert-system for an INtelligent Supply of  
Thermal Energy in Industry  
and other Large Scale Applications

# What is new in **EINSTEIN 3.0 / 3.0*plus***

The most relevant changes and new features with  
respect to the previous version *2.5plus*

# 1 Completely New User Interface based on Graphical Editors

## 1.1 Graphical System Editor

Within a fully re-designed EINSTEIN user interface, the graphical system editor, which in earlier versions has been used only for visualisation of energy flows, now is the main interface for the user in order to set-up and edit the system model (Figure 1).

Components of the system model, such as utilities or processes can be added to the model via this graphical interface. Component parameters can be read and edited by simply clicking on the corresponding symbol, and connections between the components can be established by simple mouse clicks.

A large amount of specific icons has been created in order to visually distinguish different types of equipment, piping and processes.

Furthermore, also heat exchangers are now visualised and can be edited within this graphical system editor

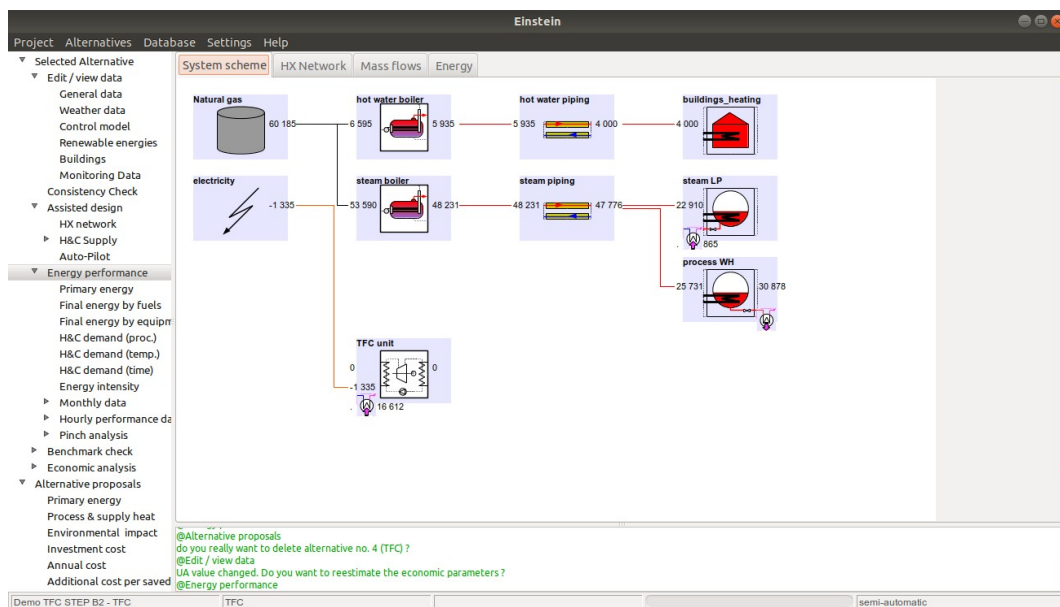


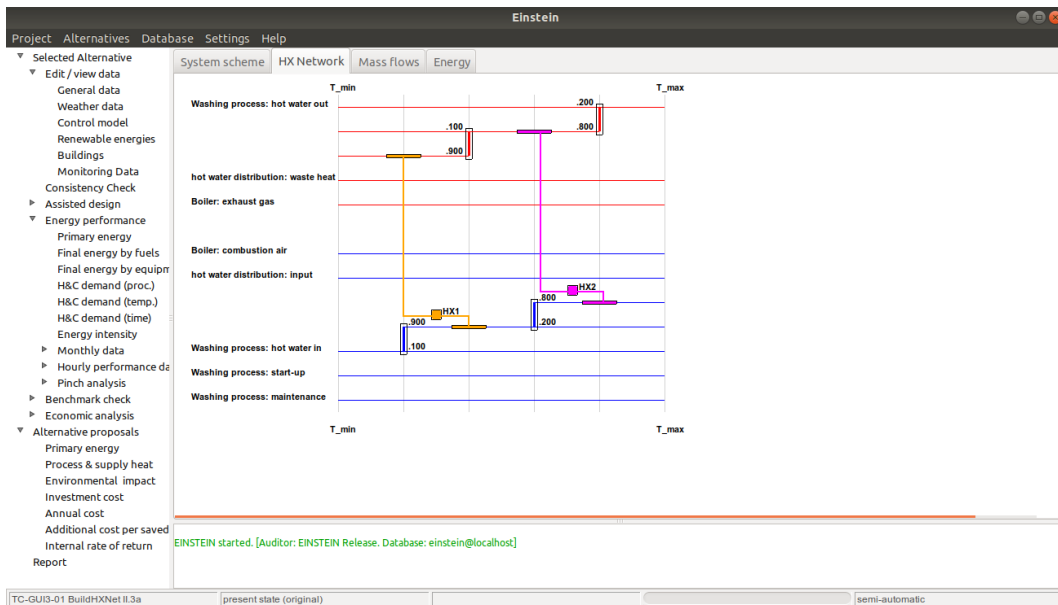
Figure 1. Scheme of a system (example).

## 1.2 New Editor for Heat Exchanger Networks and Visualisation of Hot and Cold Streams

A new editor has been added for viewing and editing heat exchanger networks. Heat exchangers can be added by simply clicking on the corresponding source and sink streams. Parameters of heat exchangers can be edited by clicking on the corresponding heat exchanger symbol

The editor offers three different types of view:

- matrix view: heat exchangers are plotted equidistant and ordered by upstream / downstream.
- linear temperature scale: heat exchangers are plotted on their place corresponding to source and sink temperatures on a linear temperature scale
- logarithmic temperature scale: as above, but using a logarithmic instead of a linear temperature scale.



**Figure 2. Editor for heat exchanger networks (matrix view).**

### 1.3 Editor for System Mass Flows (Production Flow Chart)

The mass flows in different points of the system can be correlated in different ways:

- *direct (input-output) correlation*: e.g. the output of one process may be used (fully or partially) as an input to another process
- *indirect correlation (available information  $y = f(x)$ )*: the quantity of working fluid to be processed in some process may be correlated to some other quantity in the system, e.g. it may be known from experience that for painting one car (in the average) a certain amount of solvent has to be processed in thermal post-combustion within the paintry section, etc.

EINSTEIN 3.0 disposes of a graphical editor for mass flows within the system where both types of correlations can be defined (Figure 3; solid lines are input-output connections, whereas dashed lines represent “information flows”), complemented by the corresponding back end module for the calculation and checking of mass balances.

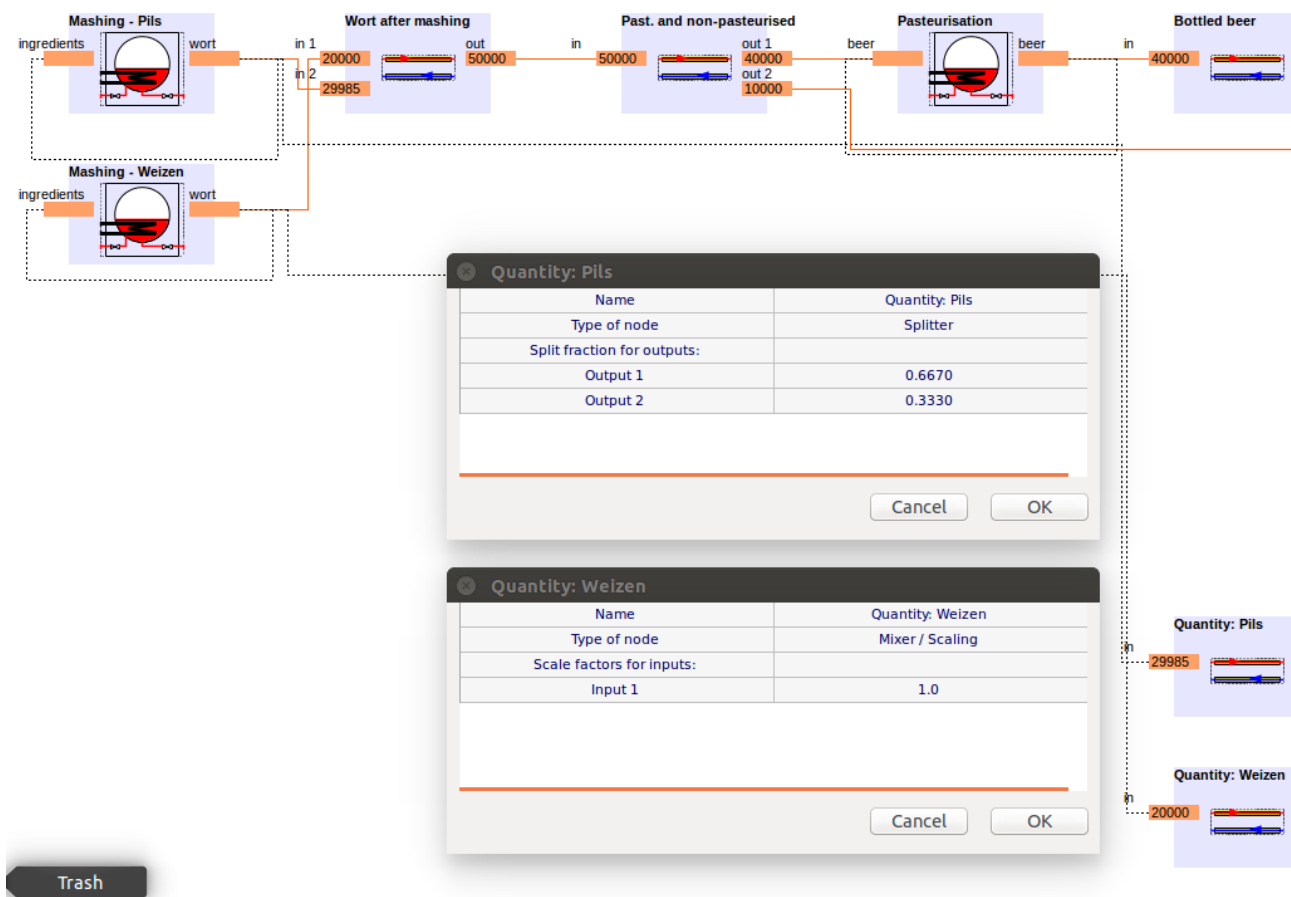


Figure 3 EINSTEIN editor for the mass flow network

Correlations between system mass flows can be used to simplify the data entry for the project. E.g. if the same medium is processed in several processing steps, the corresponding quantity or flow rate has to be defined only once.

Furthermore flow rates of some media from/to the processes can be calculated based on information on other quantities such as (final or intermediate) products or raw materials.

## 1.4 Improvement of Editors for Component Parameters and Data Bases

A special effort has been dedicated in order to make editors for component parameters (processes, equipment, piping, heat exchangers) more intuitive specially for new users of the software by rearranging parameters, and move optional parameters into a second plane in order to be easily accessible by advanced users, but be hidden at first sight in order to not confuse beginners.

### 1.4.1 Process Editor

Following the standard EINSTEIN process model (see Audit Guide, Chapter 2) process heat (or cooling) demand can be composed by 3 components:

- circulation: heating (cooling) of incoming fluids
- maintenance: compensation of heat losses and latent heat (e.g. evaporation)
- start-up: initial heating (or cooling) of reservoirs
- 

In the new process editor only the parameters corresponding to those demand components which are really used (different from 0) are visible. Other demand components can be added only if needed.

A summary table gives a quick overview of all the demand components present for a given process.

Component	Medium	Flow rate kg/h	Inlet/Initial temp. °C	Nominal power kW	Yearly energy MWh	Operating Hours h	Additional data
maintenance	-		120.50	1 600.00	20 543.90	12 840	
feed water	Water	2 617	60.00	184.28	2 366.10		view/edit
<b>TOTAL</b>					<b>22 910.01</b>		

Figure 4 New EINSTEIN process editor

### 1.4.2 Pipe and Duct Editor

Also the new pipe and duct editor follows the philosophy to make visible only those fields for data entry which are really needed. The user can preselect between a (fully) open circuit, a (fully closed) circuit and a partially closed circuit with recirculation and feed-up, and – depending on this preselection – the editor adapts showing the required fields for data entry.

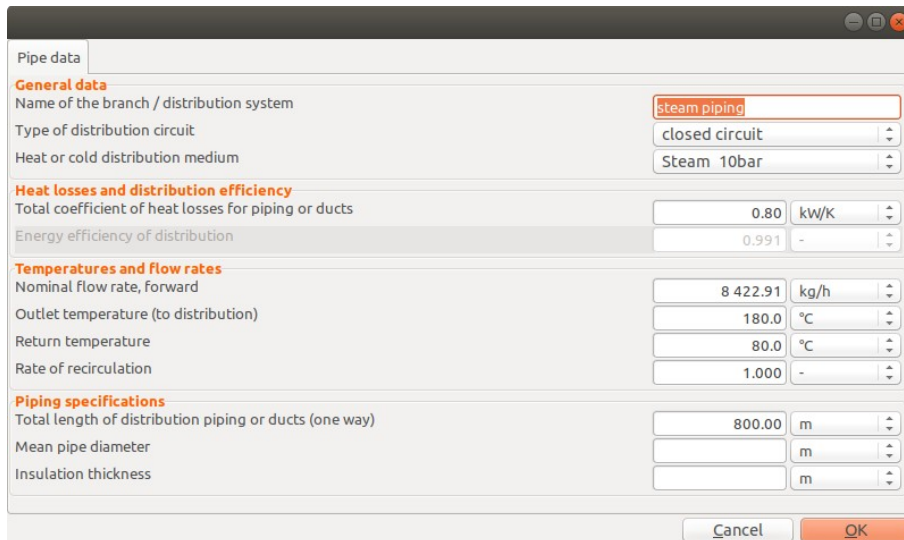


Figure 5 New EINSTein editor for ducts and pipings

### 1.4.3 Equipment Editor

Within the new equipment editor, all technical data are now grouped within a technical data table which is adapted to the specific type of equipment, so that all parameters which are not needed are hidden.

Furthermore, a clear visual distinction between nominal data and real data (average data in real operation conditions) has been introduced by locating the parameters in the corresponding column within the technical data table.

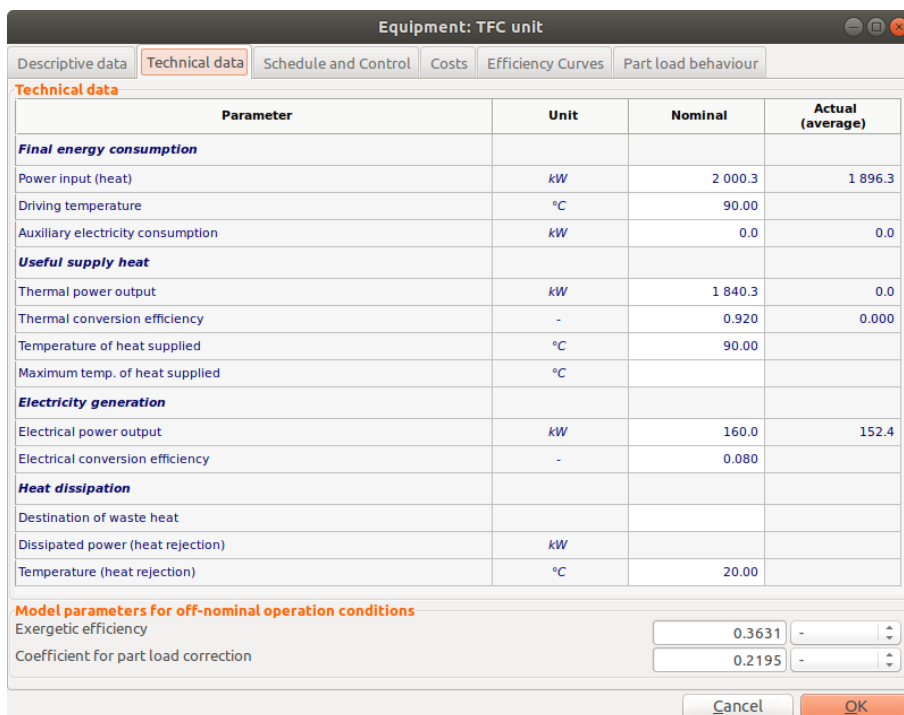


Figure 6 New EINSTein editor for equipment.

## 1.4.4 Other Component Editors

Also all other component editors (heat exchangers, other waste heat sources, etc.) have been revised following the same philosophy.

## 1.4.5 Editors for Data Bases

EINSTEIN disposes of a series of data bases which are used as background information for the calculations and can (and have to) be modified by the user. The most important data bases are:

- fluid and material properties
- fuel properties
- available equipment models (in separate data bases for different types of equipment)
- profiles (time schedules): as building complex time schedules is time consuming, the schedules can be stored and re-used in other projects

The editors for these data bases have been improved by separating the editor into an overview table, which allows to visualise the whole content of the data base (displaying some of the most relevant parameters for each entry), and an editor for a single data base entry in a separate window.

## 1.5 Output of Detailed Simulation Results

The possibility for the user to generate output of detail results of simulations has been improved.

### 1.5.1 Plot of Hourly Simulation Results in the GUI

A plot of hourly results has been added to the graphical user interface, where all energy flows within the system can be visualised (Figure 7).

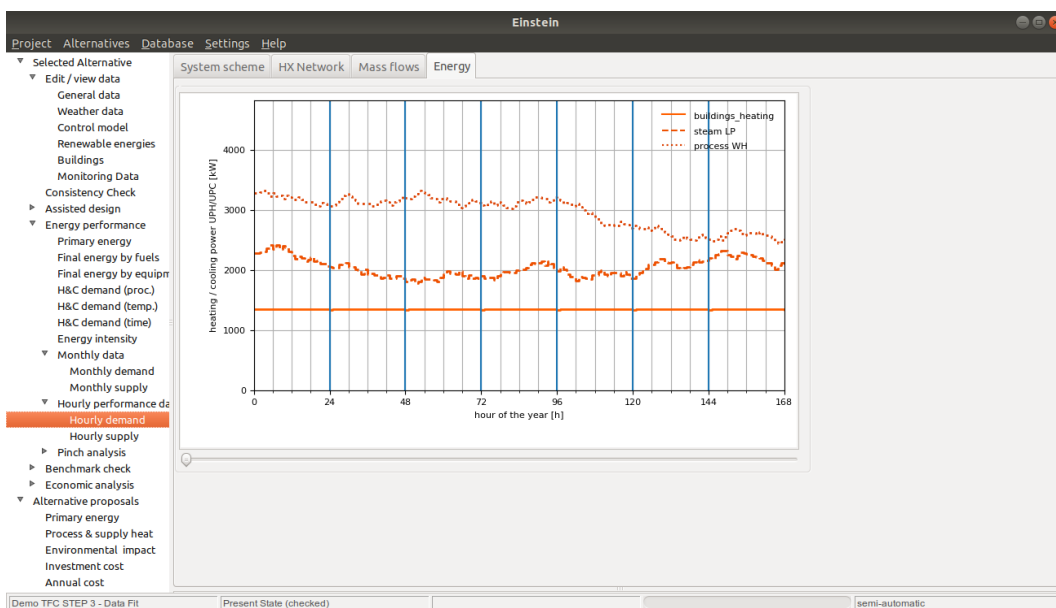


Figure 7. Hourly heating demand for the three processes as a result of data fitting. Shown for the first week (168 hours) of the year.

## 1.5.2 Data File with Hourly Simulation Results as Standard Output

The simulation engine of the tool can be configured by the user in order to generate a data file with the hourly simulation results.

## 1.6 Video Tutorials

The examples described in detail within the EINSTEIN Audit Guide Chapter 4 are pre-loaded within the EINSTEIN tool so that they can be directly opened by the user.

The **getting started** section within the user manual has been revised in order to give a better and more comprehensive description on how to use these examples for getting started with the tool.

Furthermore, some video tutorials have been elaborated and included into the “getting started” section.

An additional more complex example case has been elaborated together with a corresponding video tutorial (see the EINSTEIN user manual)

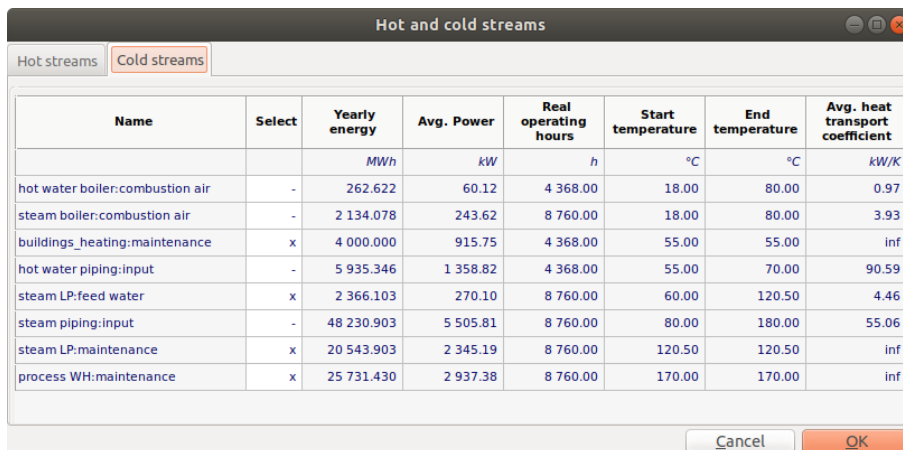


## 2 Some New Functionality

### 2.1 Improvements in Pinch Analysis Module

In the new version V3.0 of EINSTEIN it is possible to manually select / unselect a subset of hot and cold streams for pinch analysis and heat exchanger network optimisation. This gives more flexibility to pinch analysis allowing to include or exclude some of the streams:

- it allows to exclude streams for which heat recovery is not possible for some reasons, such as e.g. large distance between source and sink
- it allows to exclude (or treat separately) heat demands or waste heat streams which are only seasonally available, etc.
- it allows the user to decide whether to include within pinch analysis the heat demands at the process (process streams) or – alternatively – the corresponding heat demand of the heat supply circuit (e.g. steam piping; feed-up water or condensate return as potential cold stream).



Name	Select	Yearly energy MWh	Avg. Power kW	Real operating hours h	Start temperature °C	End temperature °C	Avg. heat transport coefficient kW/K
hot water boiler:combustion air	-	262.622	60.12	4 368.00	18.00	80.00	0.97
steam boiler:combustion air	-	2 134.078	243.62	8 760.00	18.00	80.00	3.93
buildings_heating:maintenance	x	4 000.000	915.75	4 368.00	55.00	55.00	inf
hot water piping:input	-	5 935.346	1 358.82	4 368.00	55.00	70.00	90.59
steam LP:feed water	x	2 366.103	270.10	8 760.00	60.00	120.50	4.46
steam piping:input	-	48 230.903	5 505.81	8 760.00	80.00	180.00	55.06
steam LP:maintenance	x	20 543.903	2 345.19	8 760.00	120.50	120.50	inf
process WH:maintenance	x	25 731.430	2 937.38	8 760.00	170.00	170.00	inf

Figure 8. List of cold streams with the possibility to select / unselect streams

### 2.2 Grid Integration – Variable Indexes of Electricity Grid

In EINSTEIN V3.0 it is possible to define the relevant economic and environmental parameters of the electricity grid as a function of time:

- cost (electricity tariff, separately for both purchase and sales of electricity)
- environmental parameters (primary energy ratio, CO<sub>2</sub> and nuclear waste ratio per kWh of electricity)

By this the interaction of a system with electricity grids with a high fraction of renewable energy sources with fluctuating availability (wind, PV, etc.) can be optimised.

Data can be entered via a data file containing hourly data.

### 3 Real-Time Simulation and Variable Time-Step

The capability of EINSTEIN for carrying out *real-time simulations* has been implemented. Real time simulation with EINSTEIN allows for modelling the system performance at current time and – in consequence - compare the measured real performance with simulation results. This allows for:

- adjustment (calibration) of model parameters
- detection of eventual failures in the operation of the system

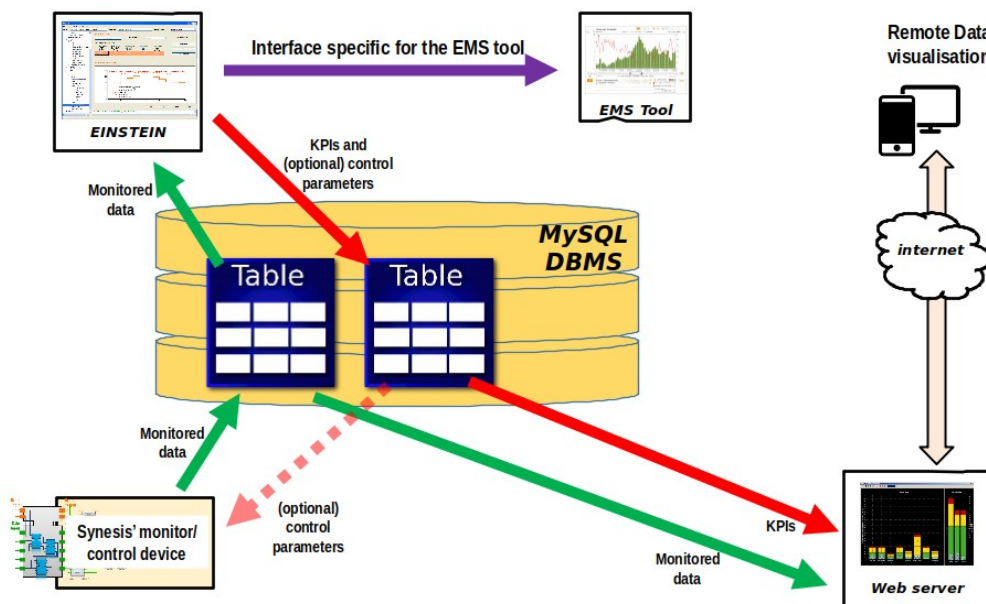
For this purpose the major developments introduced into the software have been:

#### 3.1 Protocol for Data Exchange with the System Control

A MySQL database can be used as data hub for transferring data between the system control and EINSTEIN. This allows maximum flexibility, as the data base may be located either on a local computer within the company, or on any remote server in the cloud. At the same time, also EINSTEIN can be run locally or on some remote site (Figure 9).

A protocol for the interface (structure of tables within the data base) has been defined and a default data base for testing has been set-up (see annex 1).

Data transfer can be in both regular and irregular time intervals. Typical time intervals should be in the order of few minutes for practical reasons (compromise between accuracy and volume of data to be transferred and processed).



**Figure 9.** Architecture of system control, EINSTEIN, MySQL data base and web visualisation tools. Source: Andrea Chiodi (Synesis), presentation at i-ThERM project meeting 6 in Uxbridge, London (UK), July 2018.

### 3.2 Definition of Boundary Conditions by External Data Sources

Within the simulation engine of EINSTEIN the possibility to define boundary conditions of the system model by external data sources (data files or data bases) has been added. In the current version the following parameters can be specified externally:

- process heat and cooling demands and waste heat streams (temperature, flow rates, power)
- operating temperatures (driving heat, heat rejection, ambient temperature)
- master control of equipment

Within system simulations, the parameters specified by monitoring data are considered as exogenously given and not recalculated by the simulation algorithm.

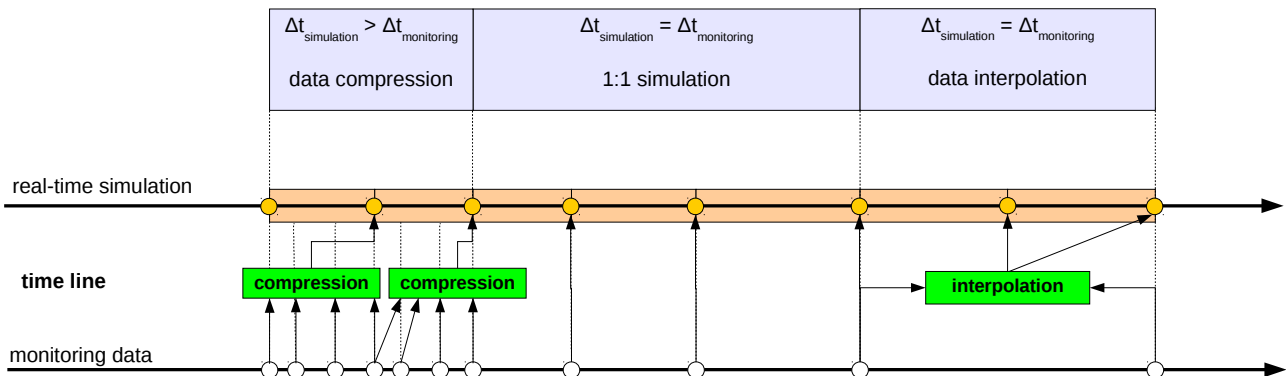
The user interface for assigning columns within data sources (data files or data bases) to internal parameters is the same as for the processing of monitoring data.

### 3.3 Simulation with Variable Time Step

EINSTEIN now is able to carry out in variable time steps well below the standard time step of 1 h. The simulation of one (or several) new time steps will be triggered by the availability of new data from the interface with the system monitoring.

The user can define limits (minimum and maximum value) for the simulation time step in EINSTEIN simulations within which the simulation time step automatically adapts to the intervals of data arriving. In principle any simulation time intervals are possible, but for practical purposes (compromise between accuracy and amount of data) simulation and measurement time steps proposed are around 0,1 h (6 minutes).

If the interval between monitoring data is higher than the maximum simulation time step, several time steps are simulated and data for the intermediate times are interpolated. In the other opposite, if monitoring data arrive with a higher frequency than the minimum simulation time step, monitoring data are compressed using sums or average values (Figure 10).



**Figure 10.** Time step in real-time simulation as a function of the intervals of availability of monitoring data.

### 3.4 Batch Execution from Command Line

For real-time simulations EINSTEIN is run in the background from the command line without opening a user interface. Simulation results are written into a data file and/or fed back into the MySQL data base and can be accessed there for visualisation or further processing.

## 4 Structural Improvement, Robustness and Testing

A series of improvements in the software structure and testing procedure have been introduced, which indirectly help to improve the user friendliness in the following senses:

- improvement of robustness of the code by modularity and a substantially increased number of standard tests
- reduction of manpower required for future updates

### 4.1 Structural Improvement of the Software Kernel

The software kernel has been gradually updated towards a clear “*model – view – controller*” structure, separating clearly the view (mainly the graphical user interface) from the calculation modules, and the latter from the data base (introduction of a database abstraction layer: all data base access is mainly carried out by a set of load and save functions for the different objects).

This evolution towards a clear software structure by itself reduces the possibility of bugs, and, furthermore, helps to increase productivity and reduce required manpower for future developments.

### 4.2 Structural Improvement of the Graphical User Interface

Taking the opportunity of the complete reorganisation of the graphical user interface, all the new and many of the existing elements of the graphical user interface have been reprogrammed based on standardised and reusable modules (objects).

The impact for the user is that, while in previous versions different GUI windows which had been written by different developers, behaved different in many ways, now almost everywhere standard widgets are used and a standard behaviour (e.g. on left/right mouse clicks etc.) of the tool in all windows is widely achieved.

Furthermore, as with the comments above on the structure on the software kernel, also within the GUI modularity helps to minimise the introduction of new and the detection of possibly existing bugs and, therefore, increase the robustness of the code.

### 4.3 Additional Test Cases for New Tool Features

For each of the new features developed during the i-ThERM project a set of standard test cases has been developed which allow for automatic testing (and if necessary bug-fixing) for each new tool release.

These test cases, although designed specially for the new features, act also as a test for many of the commonly used features of the tool and, therefore, increase also the reliability of the standard testing procedure for the standard tool features.