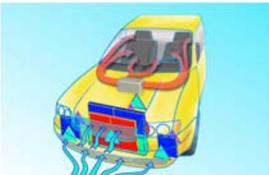
MultiObjective Optimization in Commercial Vehicle Cooling Layout

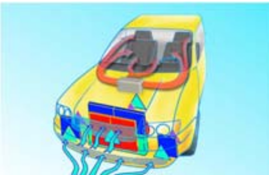
Using Genetic Algorithms to Improve Engine Cooling Performance

Nader Fateh
Esteco North America



Agenda

- Optimization Software – **modeFRONTIER**
- Optimization – single- and multi-objective
- Genetic Algorithms
- Test Case
- TRUCK Model



Software Overview

modeFRONTIER is a **m**ulti-objective **o**ptimization and **d**esign **e**nvironment, featuring:

- Process Integration (easily coupled to almost any CAE tool)
- Complete Collection of Algorithms for:
  - Design of Experiments
  - Robust Design
  - State of the art Optimization (Single- and Multiobjective)
  - Response Surface Methods
  - Data Modelling
  - Data Mining & Statistics
  - Decision Support Methods
- Java Source Enables Portability: Unix and Windows
- Network Operation (Homogeneous and Heterogeneous)

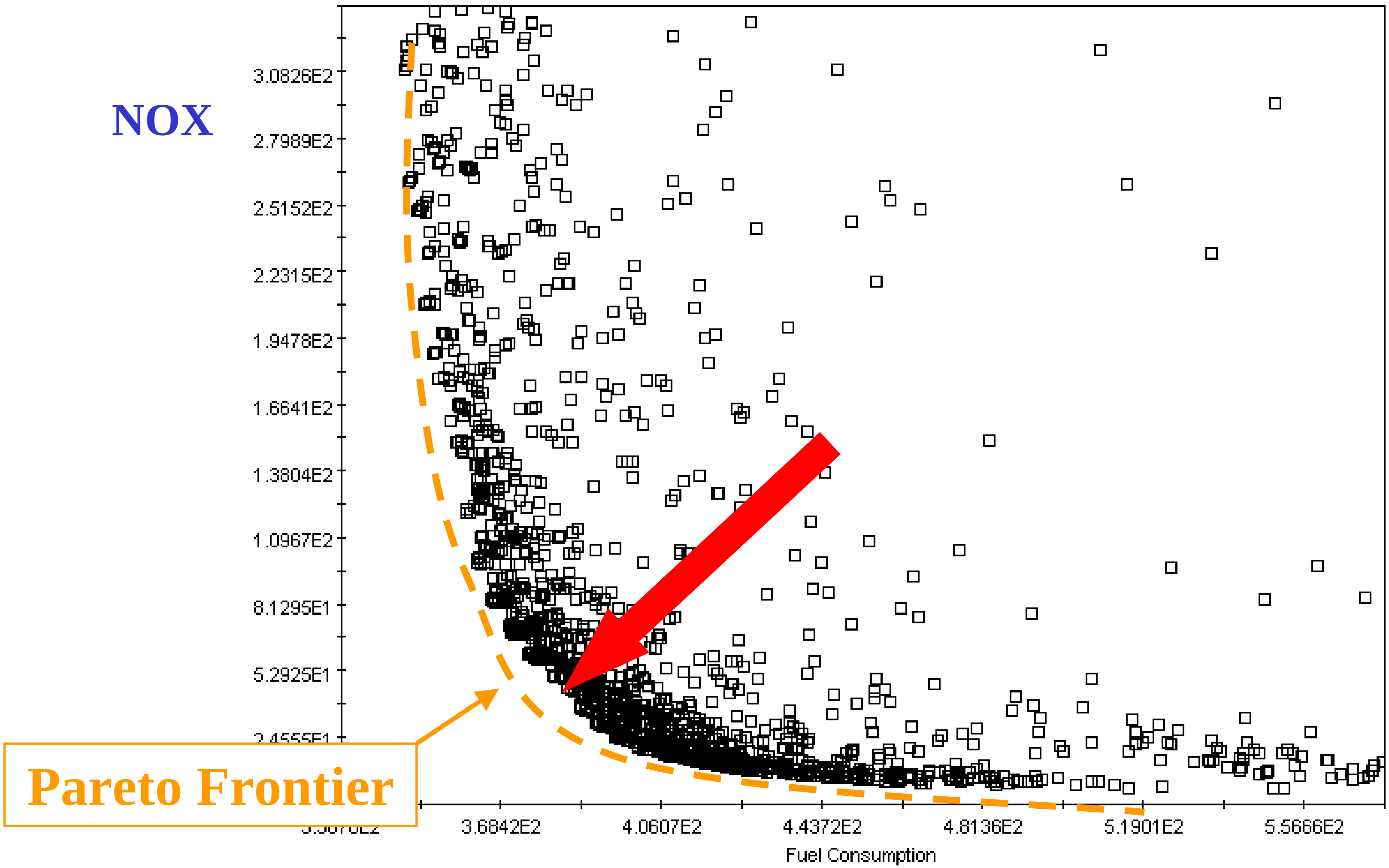
Optimization

Single-objective vs Multi-objective

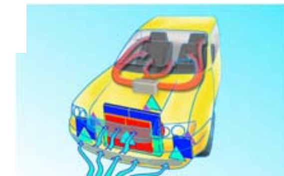
SO	MO
Typically use gradients to maximize (or minimize) a single objective function	Allows multiple objectives
Continuous Variables	Discrete variables (including "Catalog Data")
May converge to local optimum	Finds global optimum
Produces single optimum point	Produces *Pareto Frontier*, or *Trade-Off Curve*
Computationally more efficient	Requires more data points

Pareto Frontier (Trade-Off Curve)

NOX

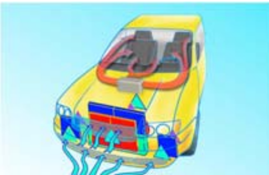


Fuel Consumption

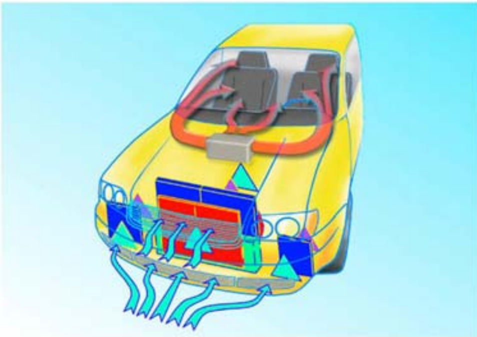


Genetic Algorithms

These mimic natural evolution, where a biological population evolves over generations to adapt to an environment through processes of selection, crossover ("reproduction") and mutation of genes.



Coupling KULI with modeFRONTIER



KULI 5.2

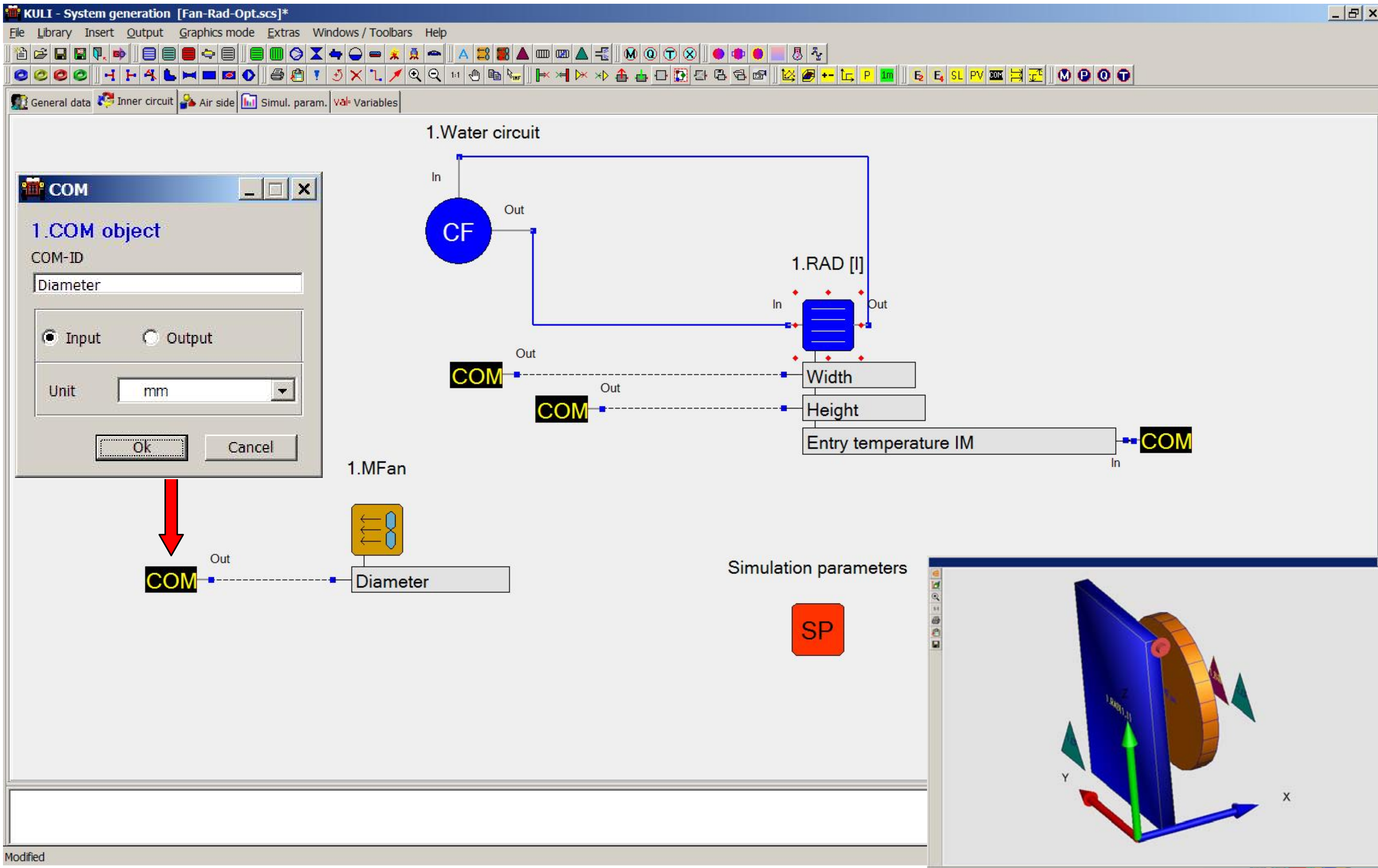
Coupling **modeFRONTIER** and **KULI**

- **KULI** can be run through Excel, using VB macros and COM objects
- **modeFRONTIER** has a direct interface to Excel, so that it can directly enter the input parameters into the Excel file, run macros, and extract the output data
- Not all variables can be modified through the COM objects **- modeFRONTIER** sets these by modifying **KULI**'s **.scs** file directly

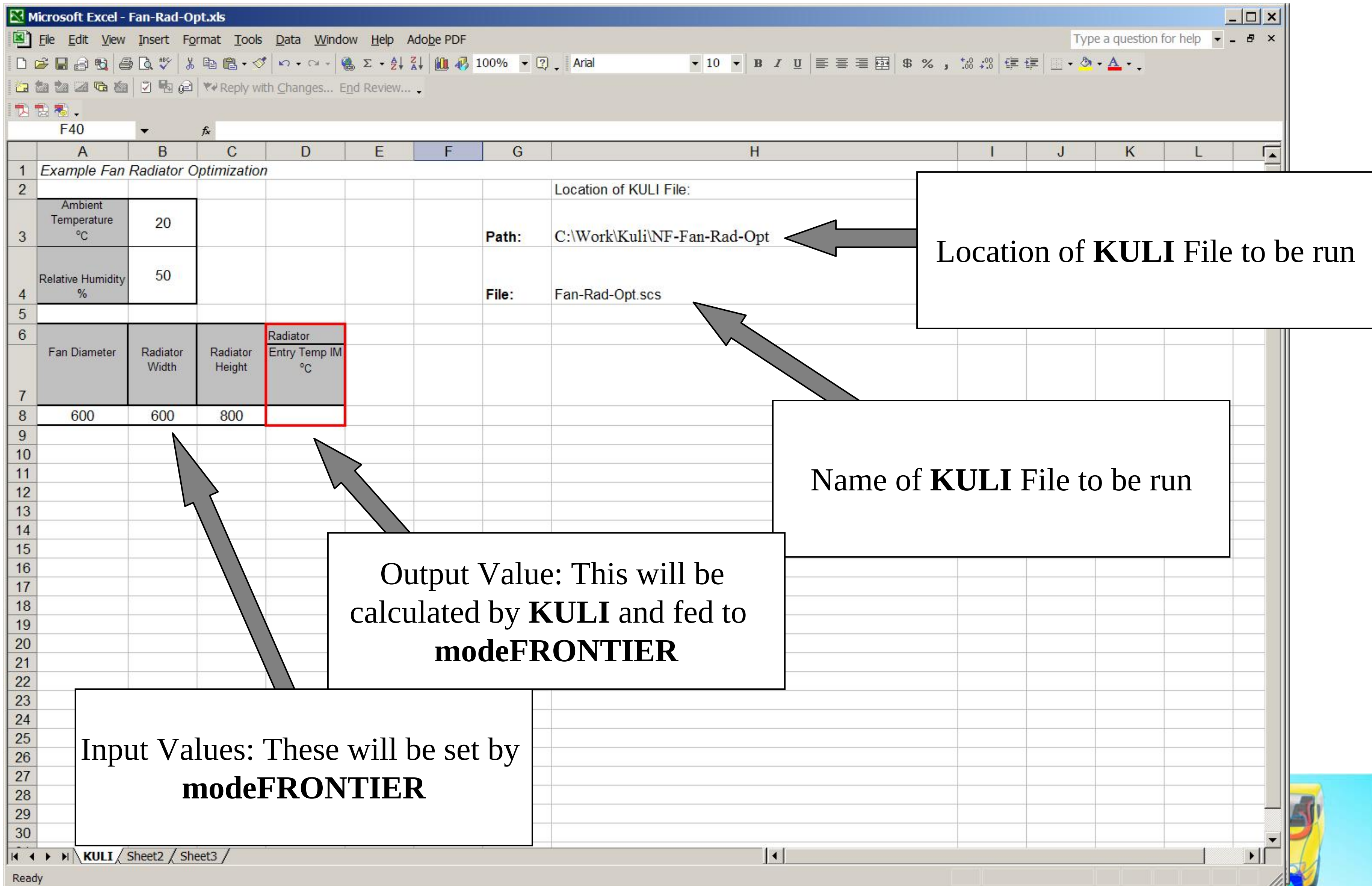
Coupling **modeFRONTIER** and **KULI** through Excel

- Simple Example:
  - – Simple system with radiator and fan
  - – 3 variables: fan diameter, radiator width and radiator height
  - – Objective: minimize radiator entry temperature
  - – As we only have one objective, we use a gradient-based algorithm (Simplex)
  - – Clearly, the program should converge to a system where all 3 variables take their maximum values

KULI Inner Circuit



KULI - Excel File

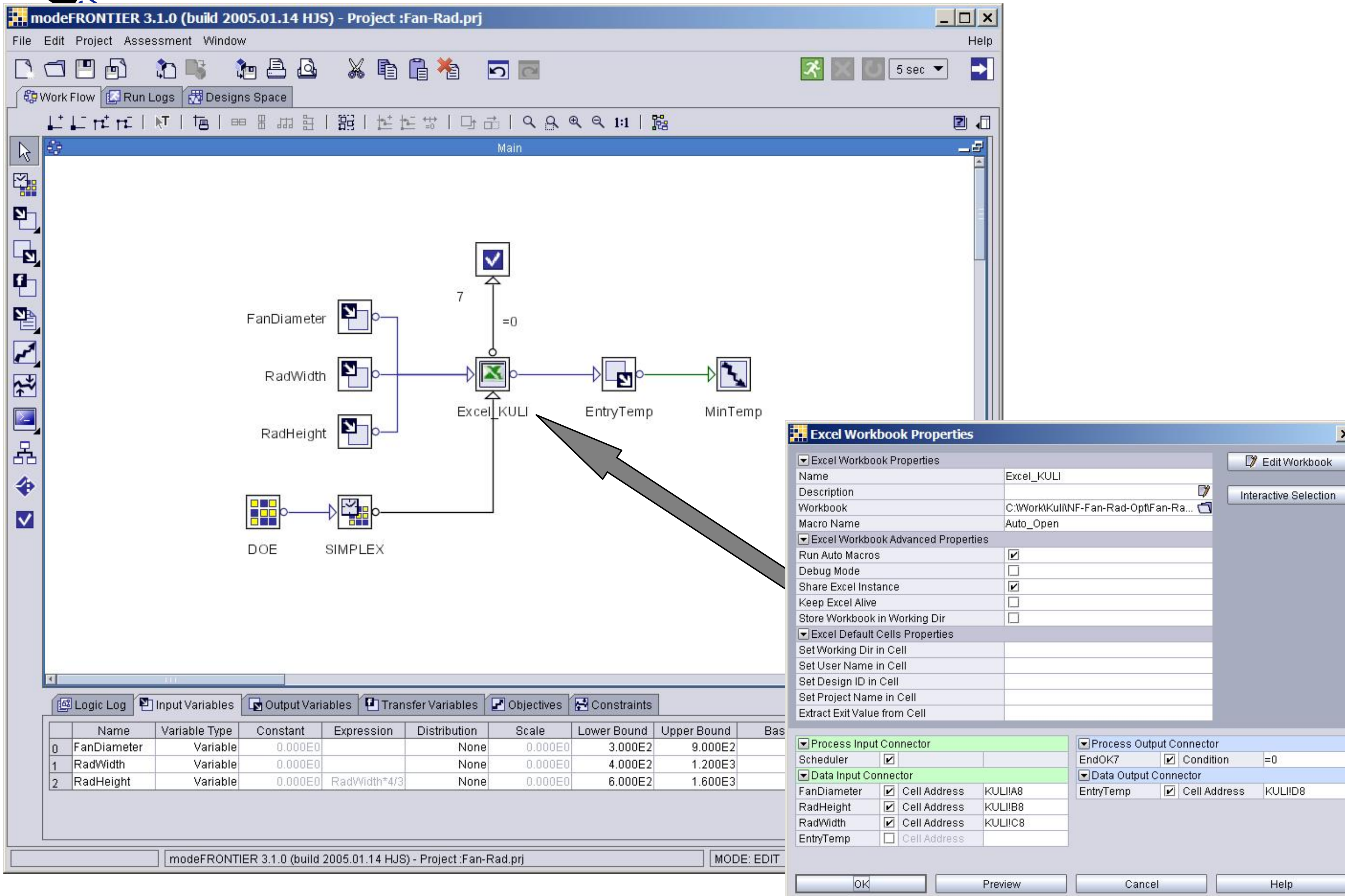


KULI - Excel VB Macro

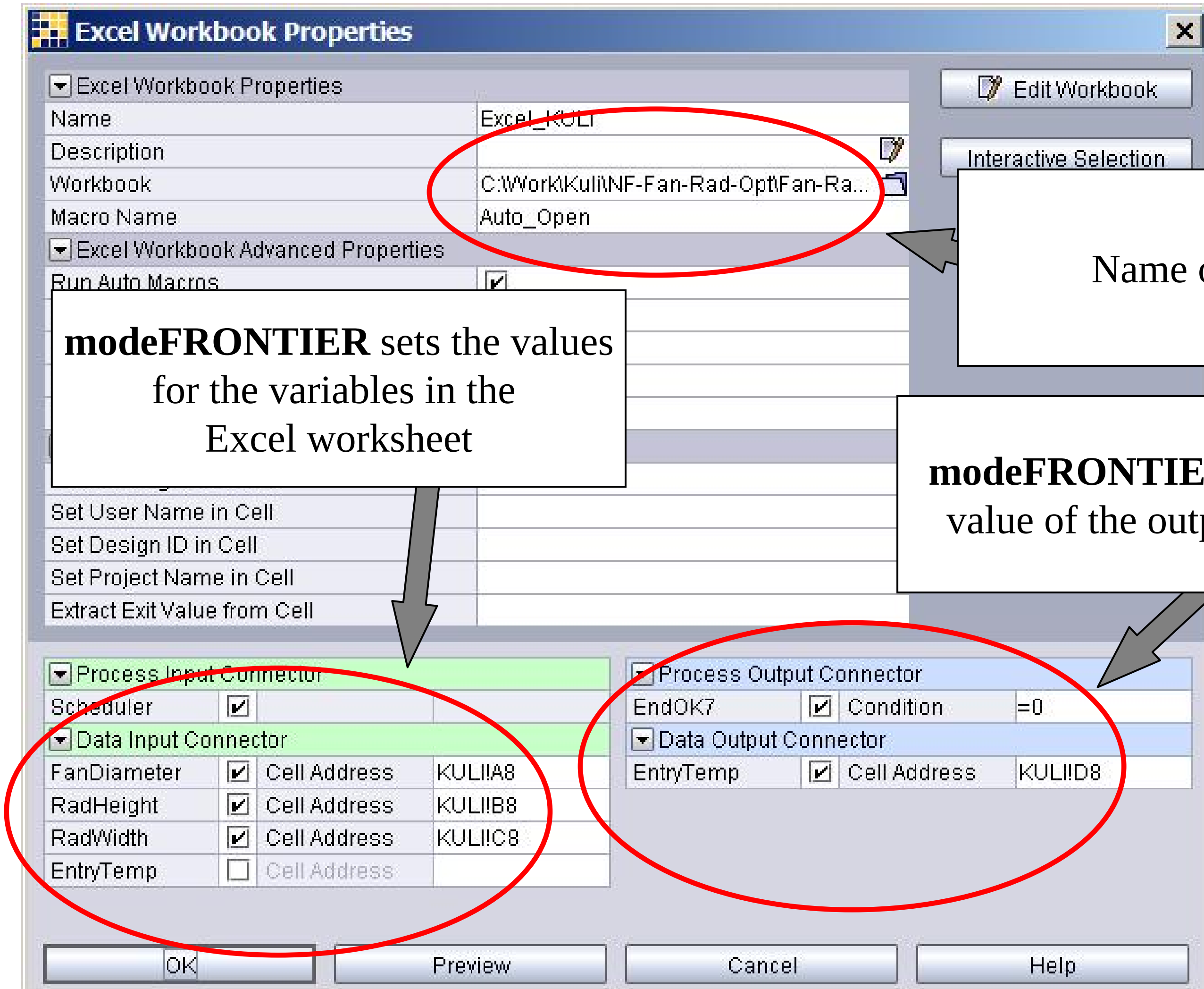
```
Sub KULIIF()
    Dim calcOK As Boolean
    Dim folder    As String
    Dim fileKULI  As String
    Dim Temp

'Getrennte Eingabe von Ordner und filename

    folder = Cells(3, 8)
    If Right(folder, 1) = "\" Or Right(folder, 1) = "/" Then folder = Left$(folder, Len(folder) - 1)
    fileKULI = folder & "\" & Cells(4, 8)

    If Dir(fileKULI, vbNormal) = "" Then  ' File exists ?

      MsgBox "File not found", , fileKULI    ' no - Error

      Exit Sub

    End If

    'set a new KULI-controller
    Set KULI = New KuliAnalysisCtr2

    'change directory name to your settings
    KULI.KuliFileName = fileKULI

    'initialize the cooling system
    calcOK = KULI.Initialize()

    calcOK = KULI.SetCOMValueByID("Diameter", Cells(8, 1))
    calcOK = KULI.SetCOMValueByID("Width", Cells(8, 2))
    calcOK = KULI.SetCOMValueByID("Height", Cells(8, 3))

    'KULI calculates the values
    KULI.SimulateOperatingPoint (1)
    Cells(8, 4) = KULI.GetCOMValueByID("EntryTempIM")

    'End KULI-analysis
    KULI.CleanUp
    Set KULI = Nothing
End Sub
```

The name used in the macro is the same as that in the COM objects

modeFRONTIER Process Flow



modeFRONTIER - Excel Node



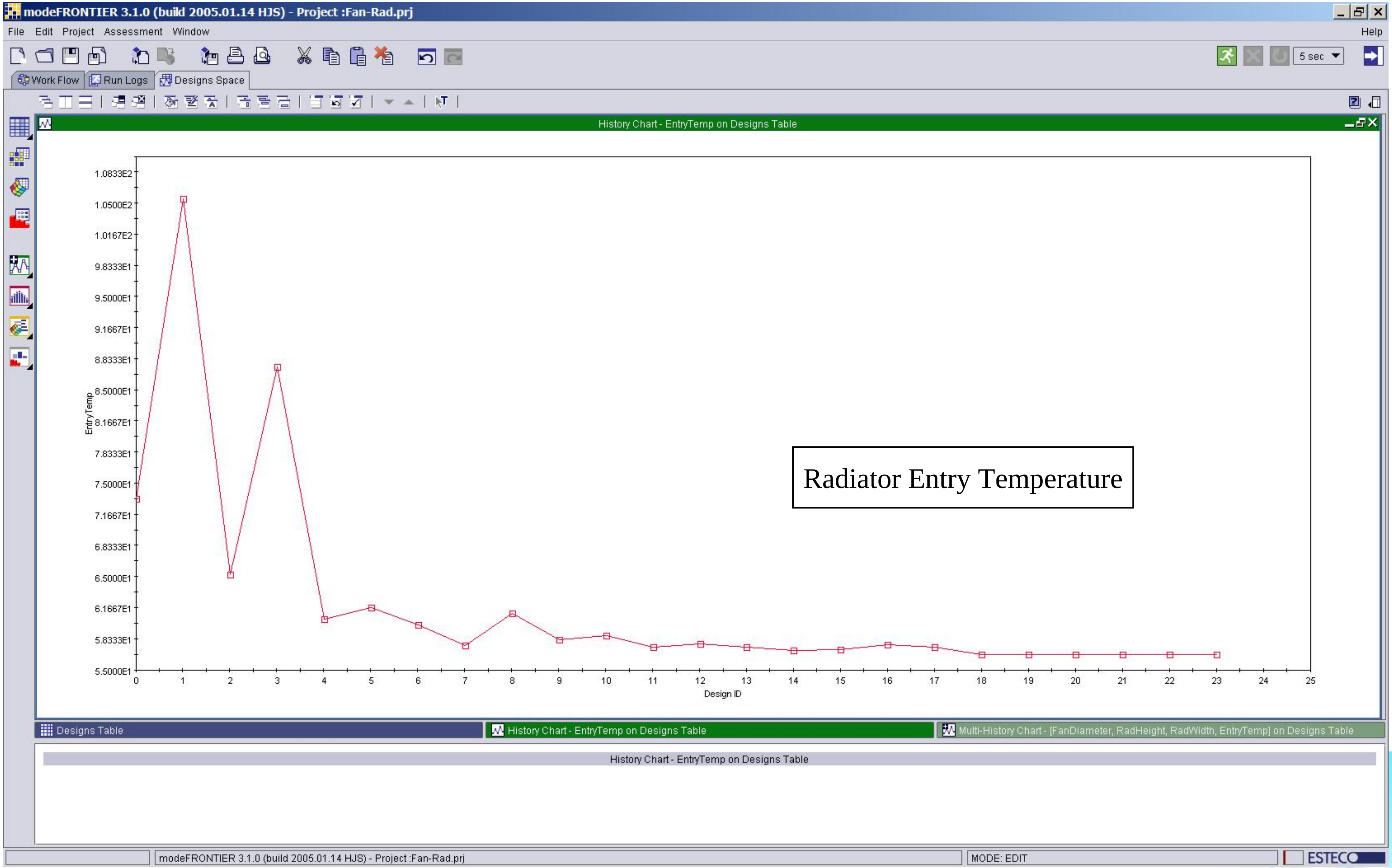
Name of Excel file

modeFRONTIER sets the values for the variables in the Excel worksheet

modeFRONTIER extracts the value of the output parameter

modeFRONTIER
Solution Development

modeFRONTIER
Radiator Entry Temperature

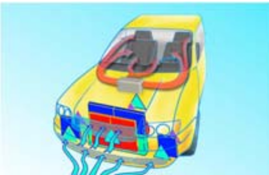


The TRUCK Model

- Given three engine ratings, with 2 operating points (peak power and peak torque) each, the goal was to maximize cooling capacity of a heat exchanger package which would pass all 6 cases.
- For each case, there was a maximum allowable Radiator Exit Temperature, and CAC Entry Temperature, given by the Engine Manufacturer’s Specifications. Therefore, one Design Iteration involved 6 KULI runs; at the end of which 6 radiator temperatures and 6 CAC temperatures were calculated. For a design to pass, all 12 calculated temperatures needed to be below the maximum allowable.

Inputs

- **modeFRONTIER** was allowed to modify the height and width of the radiator and charge air cooler, as well as the position of these components.
- Tank blockages were introduced, always with the correct position corresponding to their parent component.
- The input variables were constrained to ensure that all components remained within a predefined envelope.
- Constraints were also placed on the width and height of each component, to prevent searches in parts of design space which would clearly not produce solutions of interest.



The Objectives

- In order to try to maximize the margin by which the 6 cases passed, two 2 quantities, $\mathbf{T_{Rtot}}$ and $\mathbf{T_{CACtot}}$ were calculated within **modeFRONTIER**:

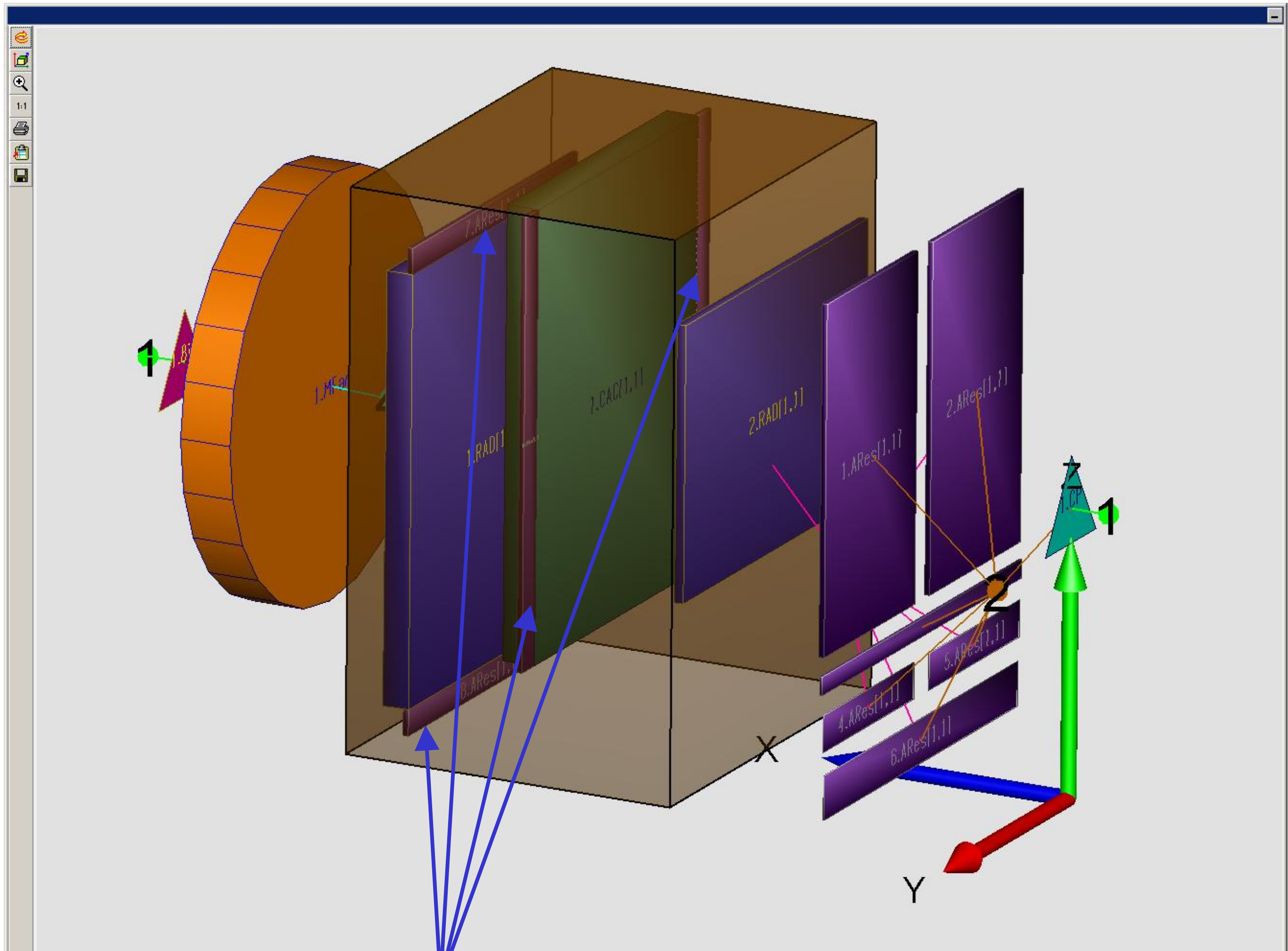
$$T_{Rtot} = \sum_{n=1}^{6} (T_{Rnmax} - T_{Rn})$$

$$T_{CACtot} = \sum_{n=1}^{6} (T_{CACnmax} - T_{CACn})$$

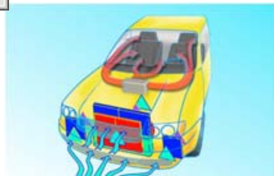
  where $\mathbf{T_{Rnmax}}$ and $\mathbf{T_{CACnmax}}$ are the maximum allowable temperatures for radiator and CAC respectively, for case n.

- Two objectives were defined: to maximize each of $\mathbf{T_{Rtot}}$ and $\mathbf{T_{CACtot}}$

KULI Model



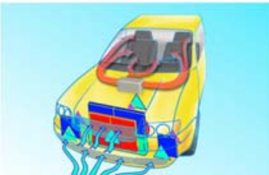
Tank Blockages



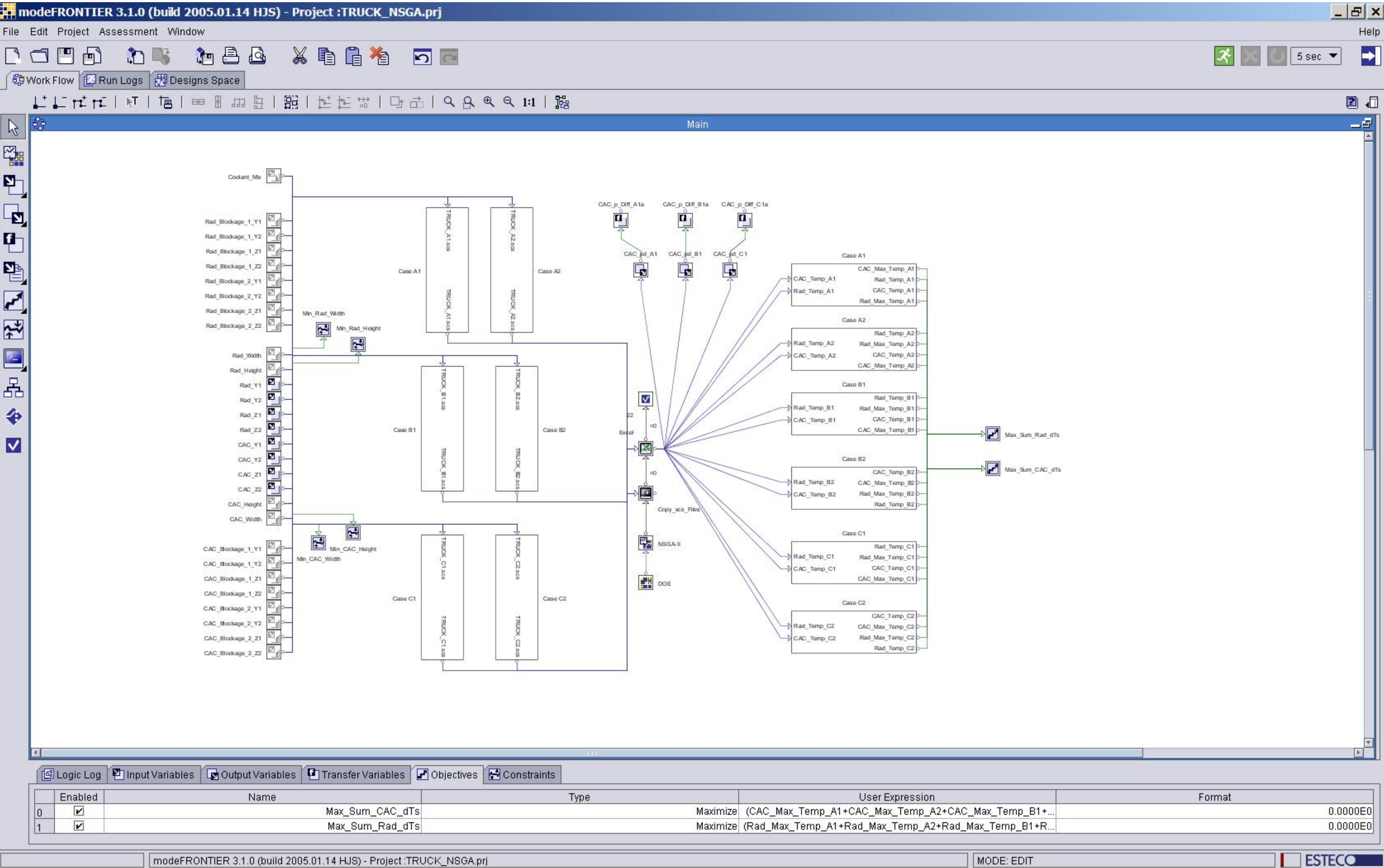
Optimization Parameters

8 Independent Variables:

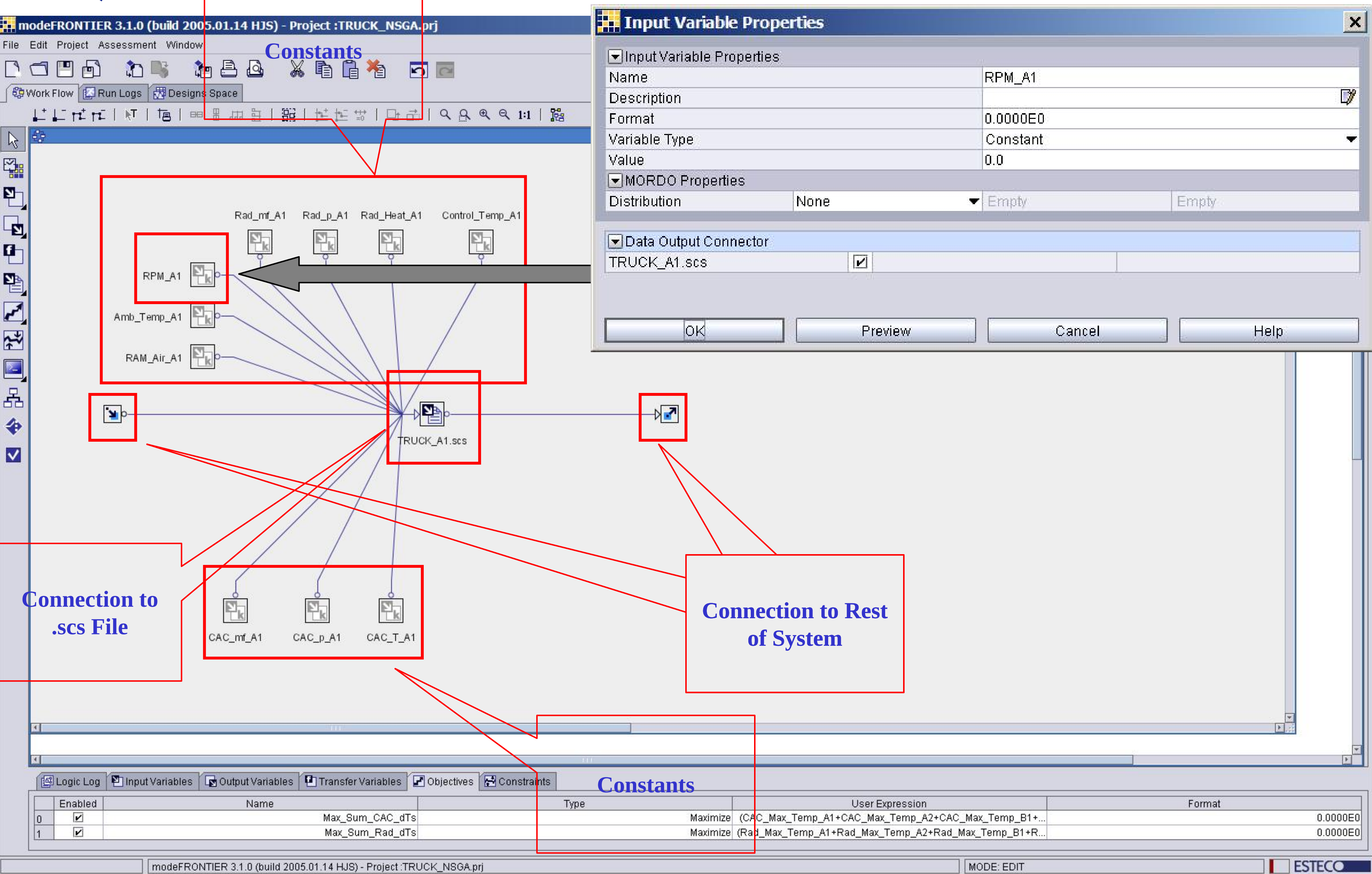
Variable	Minimum	Maximum	Step (mm)
Radiator	$Y1_{min}$	$Y1_{max}$	5
Radiator	$Y2_{min}$	$Y2_{max}$	5
Radiator	$Z1_{min}$	$Z1_{max}$	5
Radiator	$Z2_{min}$	$Z2_{max}$	5
CAC	$Y1_{min}$	$Y1_{max}$	5
CAC	$Y2_{min}$	$Y2_{max}$	5
CAC	$Z1_{min}$	$Z1_{max}$	5
CAC	$Z2_{min}$	$Z2_{max}$	5



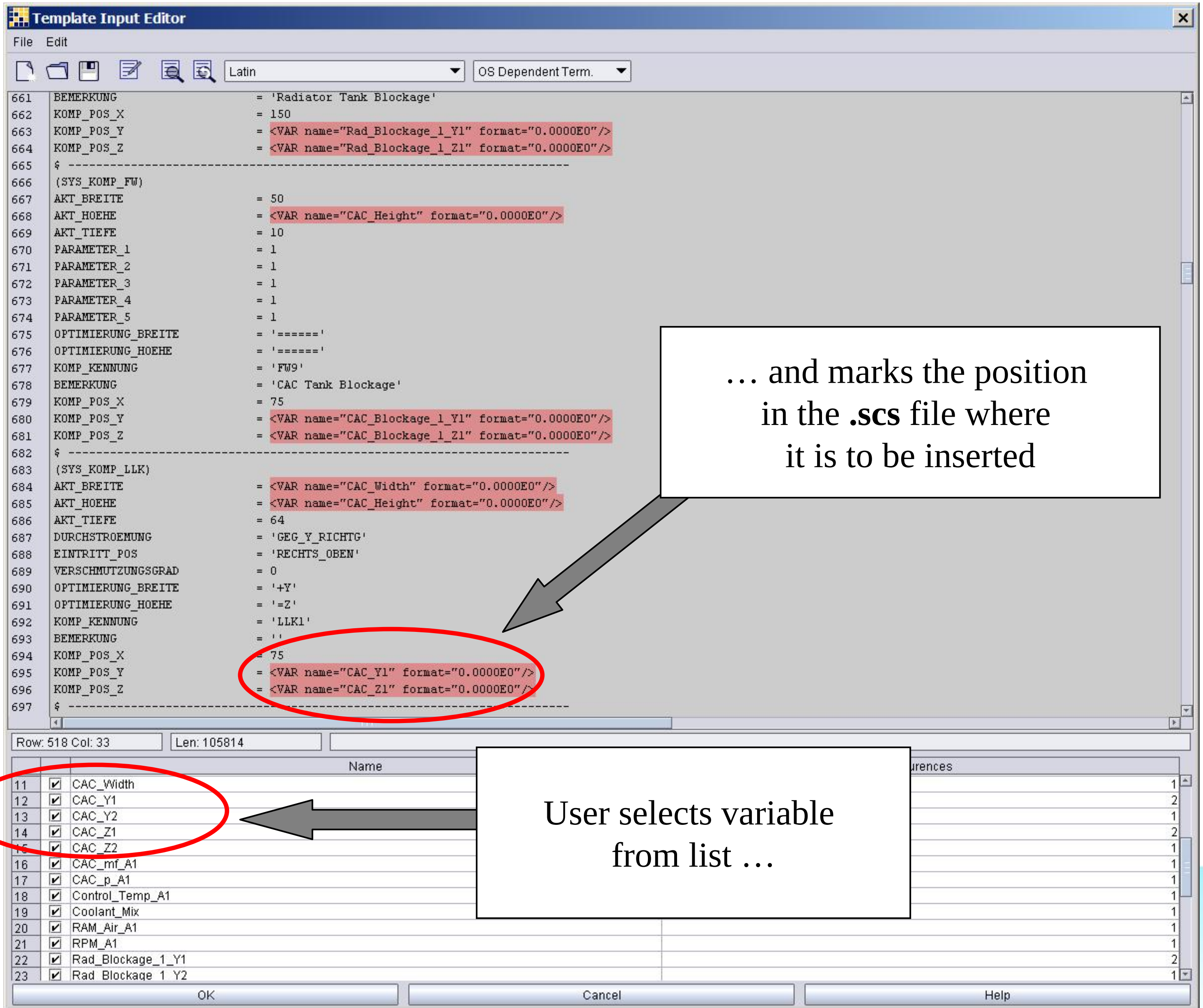
modeFRONTIER Work Flow



modeFRONTIER - Input Subsystem



Integration:
modeFRONTIER modifies the **.scs** file



Integration
Excel File Interface

Microsoft Excel - PACCAR_R_6.xls

Excel Sheet for TRUCK

	Radiator Entry	CAC Exit Temperature	CAC dP	Directory	File
Case A1:	0	0	0	C:\Work\Frontier\KULI\TRUCK	TRUCK_A1.scs
Case A2:	0	0	0	C:\Work\Frontier\KULI\TRUCK	TRUCK_A2.scs
Case B1:	0	0	0	C:\Work\Frontier\KULI\TRUCK	TRUCK_B1.scs
Case B2:	0	0	0	C:\Work\Frontier\KULI\TRUCK	TRUCK_B2.scs
Case C1:	0	0	0	C:\Work\Frontier\KULI\TRUCK	TRUCK_C1.scs
Case C2:	0	0	0	C:\Work\Frontier\KULI\TRUCK	TRUCK_C2.scs

KULI / Sheet2 / Sheet3

For each case the Excel file passes the location of the file, and the file name to the VB macro.
Excel is also used to extract the 2 temperatures and the dp across the CAC

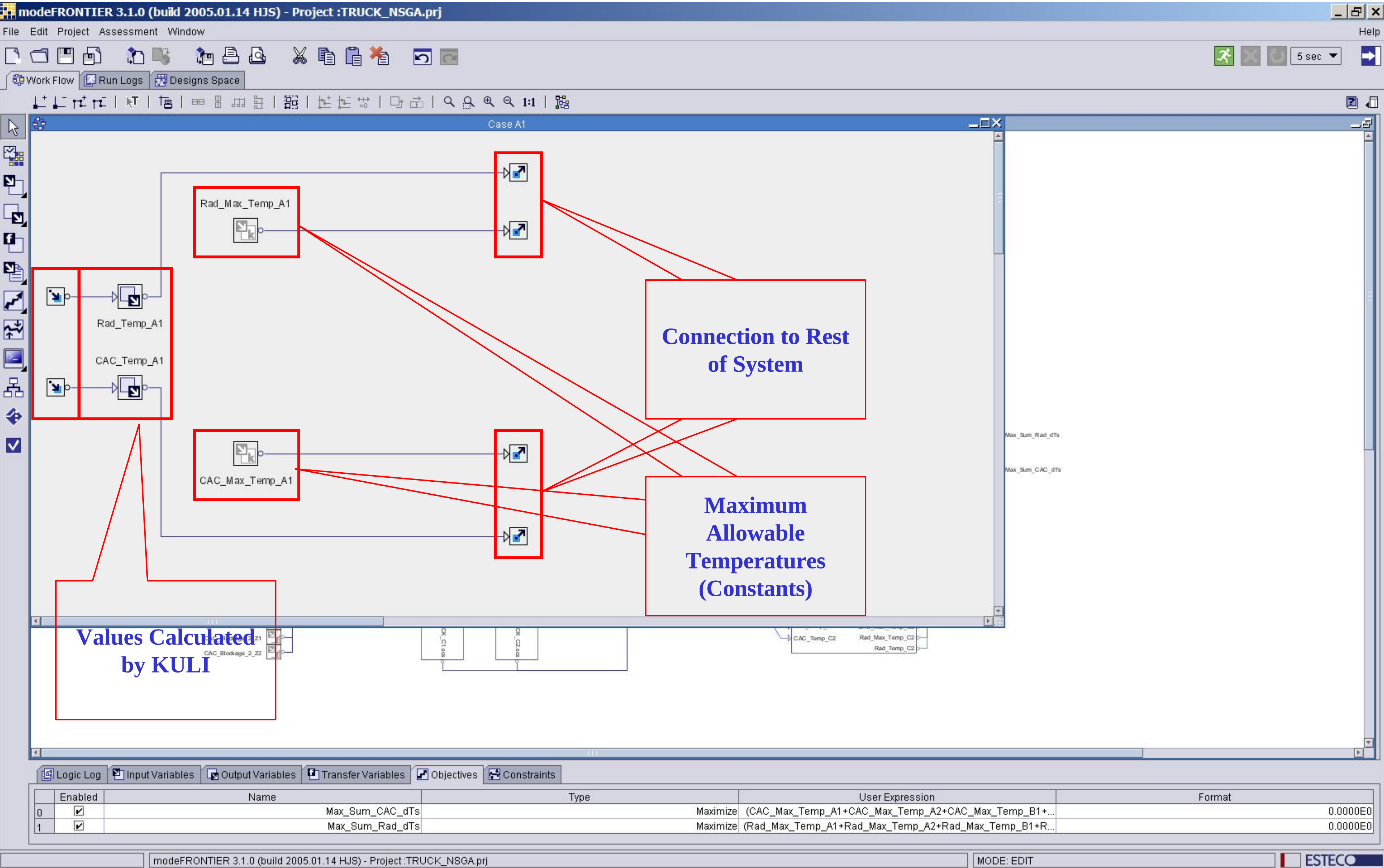
Integration
Running KULI in Batch Mode

```
Sub Auto_Open()
Application.DisplayAlerts = False
    KULIA1
    KULIA2
    KULIB1
    KULIB2
    KULIC1
    KULIC2
End Sub
```

KULI runs 6 times every time the Excel file is opened

```
Sub KULIA1()
    Dim calcOK As Boolean
    Dim folder    As String
    Dim fileKULI  As String
    Dim Temp
'Getrennte Eingabe von Ordner und filename

    folder = Cells(4, 5)
    If Right(folder, 1) = "\" Or Right(folder, 1) = "/" Then folder = Left$(folder, Len(folder) - 1)
    fileKULI = folder & "\" & Cells(4, 6)

    If Dir(fileKULI, vbNormal) = "" Then  ' File exists ?

      MsgBox "File not found", , fileKULI     ' no - Error

      Exit Sub

    End If

    'set a new KULI-controller
    Set KULI = New KuliAnalysisCtr2
    'change directory name to your settings
    KULI.KuliFileName = fileKULI
    'initialize the cooling system
    calcOK = KULI.Initialize()

    'KULI calculates the values
    KULI.SimulateOperatingPoint (1)

    Cells(4, 2) = KULI.GetCOMValueByID("EntryTempIM")
    Cells(4, 3) = KULI.GetCOMValueByID("ExitTempIM")
    Cells(4, 4) = KULI.GetCOMValueByID("PressureDifferenceIM")

    'End KULI-analysis
    KULI.CleanUp
    Set KULI = Nothing
```

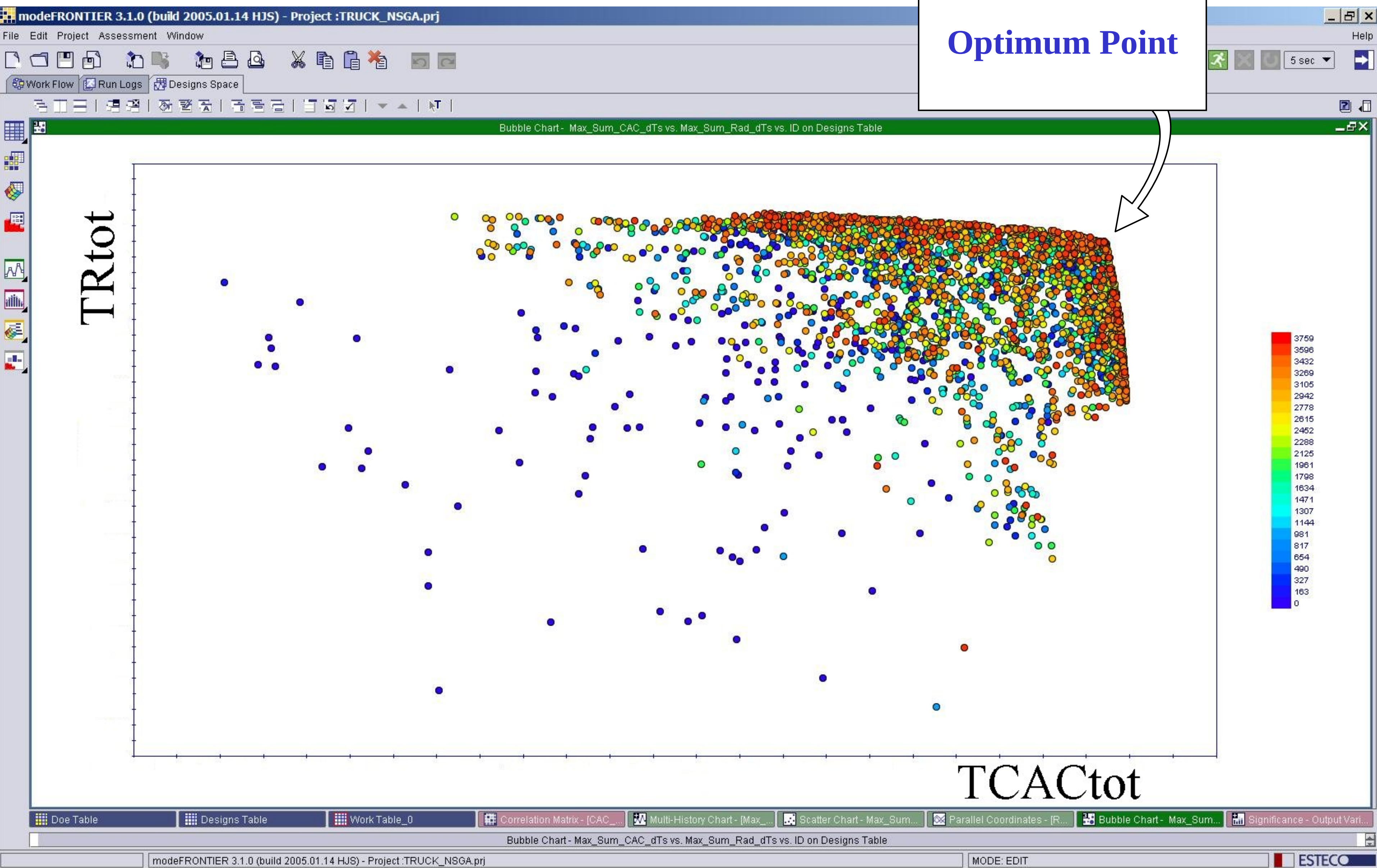
modeFRONTIER - Output Subsystem



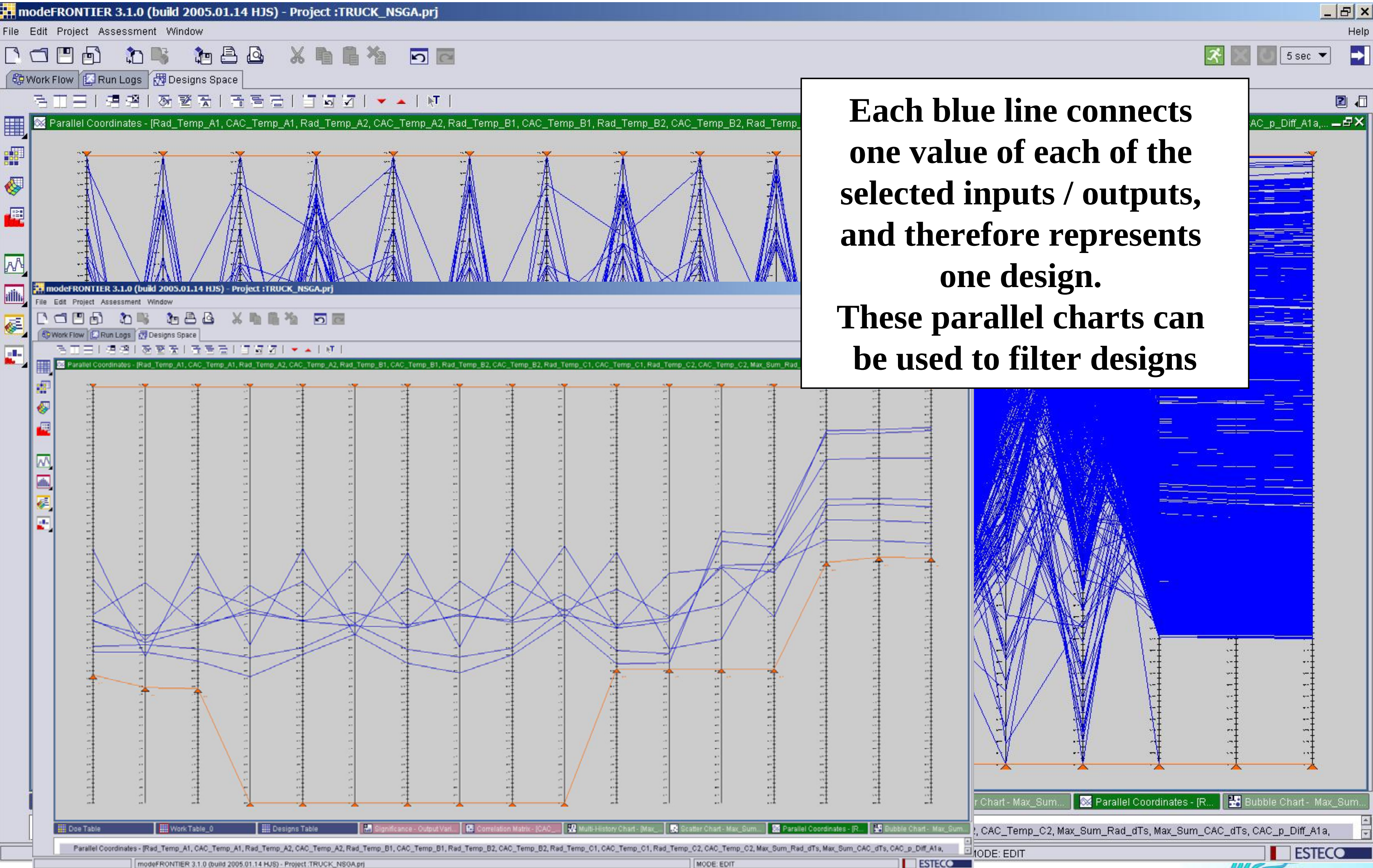
Run Statistics

- Population Size: 32
- Number of Generations: 200
- Total Number of Simulations: 6400
  (many repeated designs)
- Algorithm: NSGA-II
  (Genetic Algorithm)
- Time: Approx 15 hours on
  DELL Inspiron 8500, 2.4 GHZ

Results – Scatter Chart

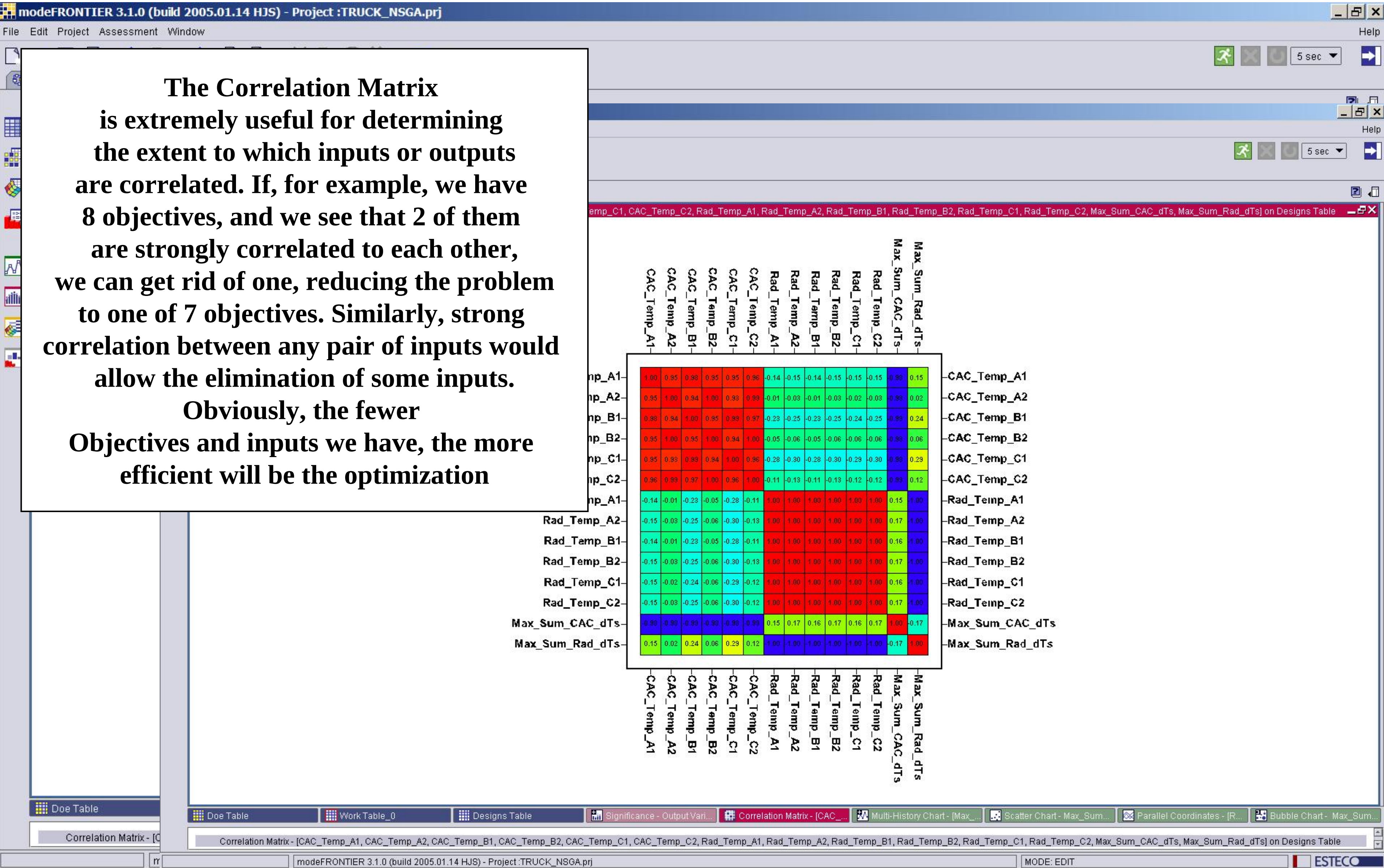


Postprocessing - Parallel Charts



Each blue line connects one value of each of the selected inputs / outputs, and therefore represents one design.
These parallel charts can be used to filter designs

Postprocessing – Correlation Matrix



The Correlation Matrix is extremely useful for determining the extent to which inputs or outputs are correlated. If, for example, we have 8 objectives, and we see that 2 of them are strongly correlated to each other, we can get rid of one, reducing the problem to one of 7 objectives. Similarly, strong correlation between any pair of inputs would allow the elimination of some inputs. Obviously, the fewer Objectives and inputs we have, the more efficient will be the optimization

General Remarks

- The engine studied biased heat rejection to the radiator. modeFrontier recognized this and developed a stacked system that balanced the cores to the engine.
- Generally, at the completion of the optimization process, the engineer / designer is presented with several Pareto Frontiers (Trade-Off Curves) – one for every set of 2 objectives.
- In this case there was only a single Pareto Frontier (as we only had 2 objectives), and the optimal point was easy to select

General Remarks (contd)

- The choice of one design over another is usually associated with an improvement in at least one of the goals at the expense of at least one of the others. This is the trade-off.
- Once a design has been chosen, a robust design analysis can be performed (using the MORDO module in modeFrontier) to ensure that the performance does not deteriorate rapidly in the case where there are small changes in input parameters, operating conditions, etc.

www.esteco.com