FME HighEFF

Centre for an Energy Efficient and Competitive Industry for the Future

Deliverable D3.1_2017_04

Numerical framework for power cycle simulation and optimization

Delivery date: 2017-11-30

Organisation name of lead beneficiary for this deliverable:

SINTEF Energy Research

HighEFF- Centre for an Energy Efficient and Competitive Industry for the Future is one of Norway's Centre for Environment-friendly Energy Research (FME). Project co-funded by the Research Council of Norway and Industry partners. Host institution is SINTEF Energi AS.		
Dissemination Level		
PU	Public	X
RE	Restricted to a group specified by the consortium	

Deliverable number:	D3.1_2017.04
ISBN number:	-
Deliverable title:	Numerical framework for power cycle simulation and optimization - FlexCS
Work package:	WP3.1
Deliverable type:	Presentation
Lead participant:	Brede Hagen

Quality Assurance, status of deliverable		
Action	Performed by	Date
Verified (WP leader)	Trond Andresen	30.11.2017
Reviewed (RA leader)	Trond Andresen	30.11.2017
Approved (dependent on nature of deliverable)*)	Trond Andresen	08.12.2017

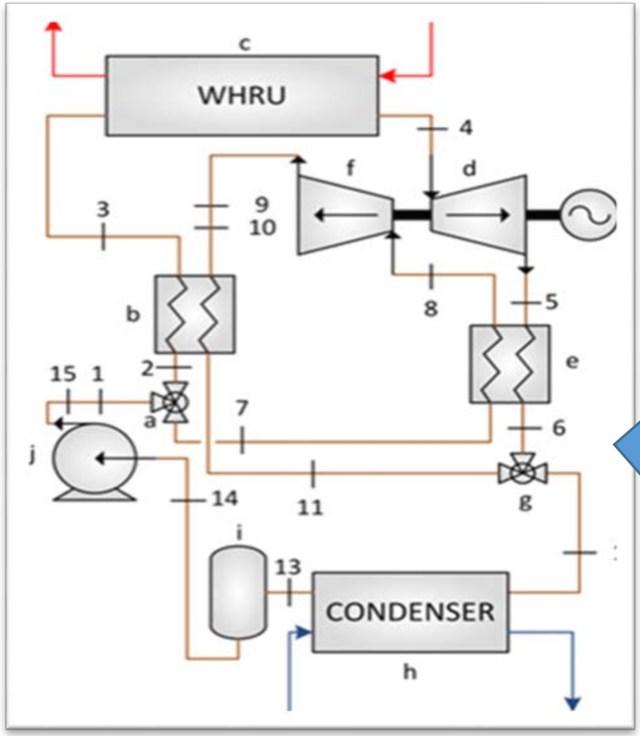
*) *The quality assurance and approval of HighEFF deliverables and publications have to follow the established procedure. The procedure can be found in the HighEFF eRoom in the folder "Administrative > Procedures".*

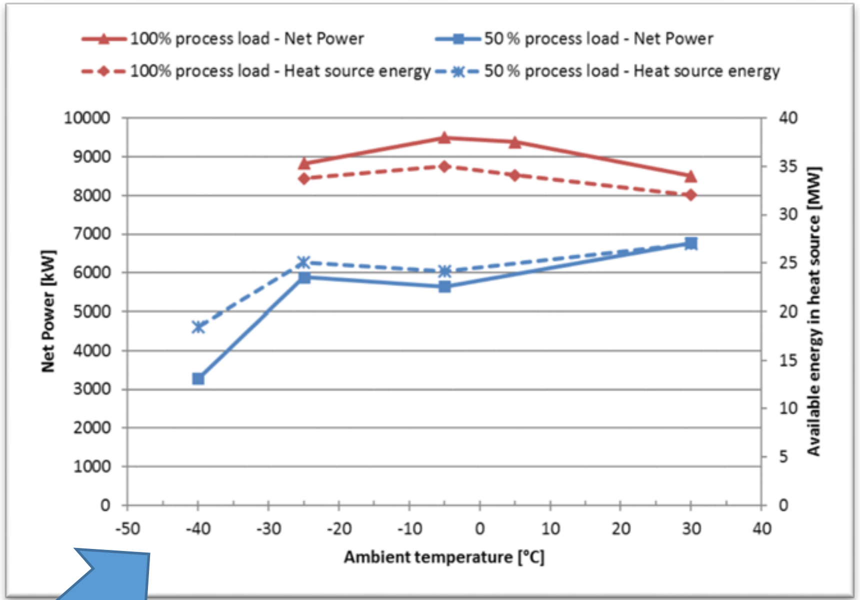
Authors		
Author(s) Name	Organisation	E-mail address
Brede Hagen	SINTEF Energy Research	Brede.hagen@sintef.no
Trond Andresen	SINTEF Energy Research	Trond.andresen@sintef.no

Abstract

Numerical framework for power cycle simulation and optimization - FlexCS

```
'define NCOMP 11

.nt btcycle () {

  FCSSystem *system = FCSSystem_create();
  initSolver(system);

  system->nrefstreams = btcycleInitInletStreams(system);
  initComponents (system);
  FCSSystem_simpleReport (system);
  initVariables(system);
  FCSSystem_initLogFiles(system);

  solveSystem(system);

  if (!offDesignExp.on) {
    FCSExpander_SaveExpanderDesign (system->expanderA->buf[0]);
    FCSExpander_SaveExpanderDesign (system->expanderA->buf[1]);
  }

  FILE *systemSimpleReport;
  FILE *systemReport;
  systemSimpleReport = fopen("systemstreamresults.res","w");
  systemReport = fopen("systemreport.res","w");
  FCSSystem_writeSimpleReport (system, systemSimpleReport);
  FCSSystem_writeFullReport (system, systemReport);
  fclose(systemSimpleReport);
  fclose(systemReport);

  if (tabulatesource.on) {
    int istream;
    istream = FCSSystem_getInletIndex (system, "sourceinlet");
    if (istream<0) AAAUtil_panic ("destroy: sourceInlet not found");
    FCSStream *sourceInlet = system->streamA->buf[istream];
    FHXThermotab_destroy(sourceInlet->node->thermotab);
    sourceInlet->node->thermotab == AAA_NULL;
  }
  FCSSystem_destroy(system);

  AAAMemory_info ();
```

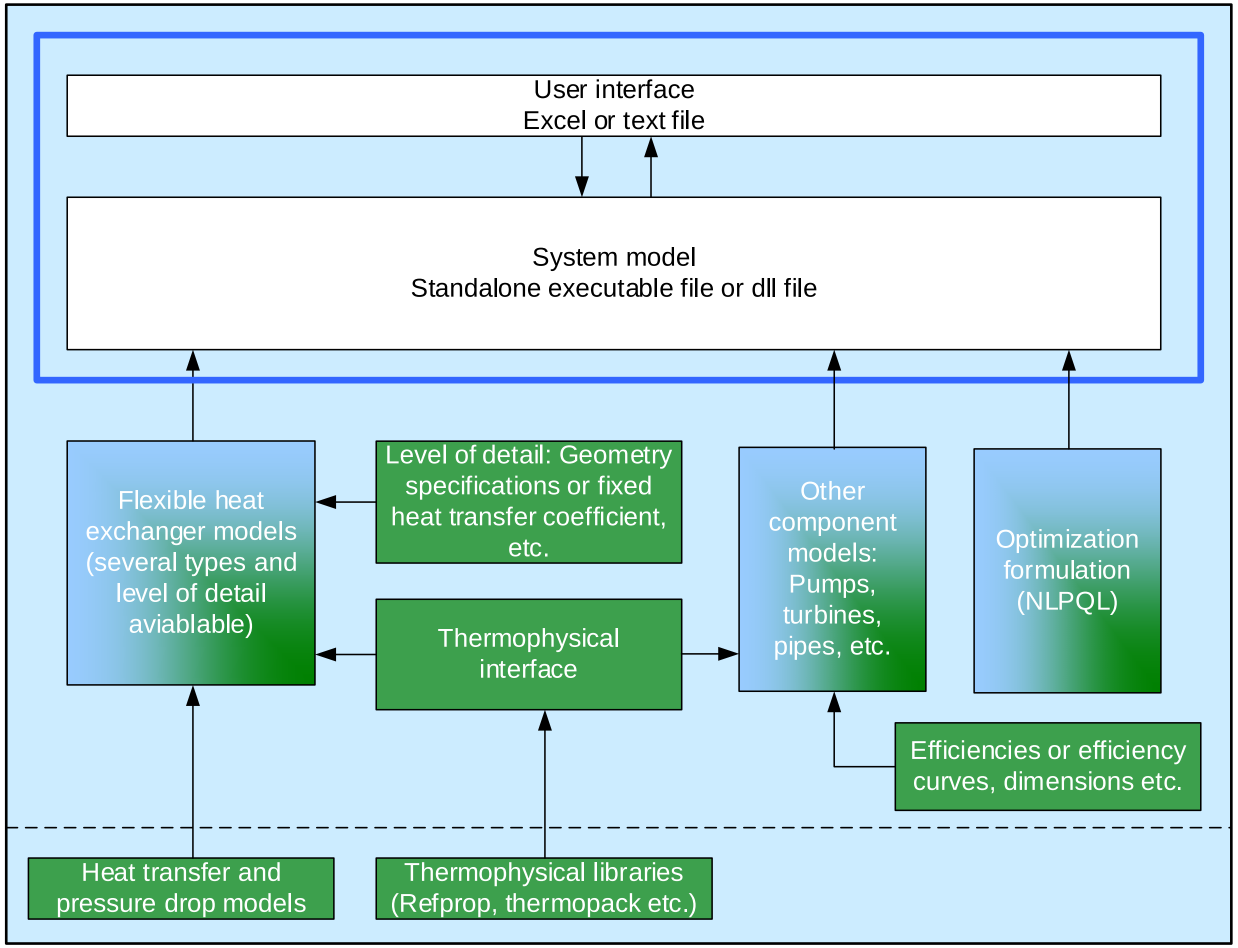




The numerical framework:
Flexible cycle simulator (FlexCS)

What is FlexCS?

- In-house "Cycle simulator" with flexible building blocks/component models
- Access to fluid properties, correlations for heat transfer and pressure drop and numerical methods
- Medium-high user threshold to assemble models (code written in C)
- Low-Medium user threshold to run parametrized models (user interface is text file or Excel)
- Cycle model can choose HX-models of different levels of detail according to the task

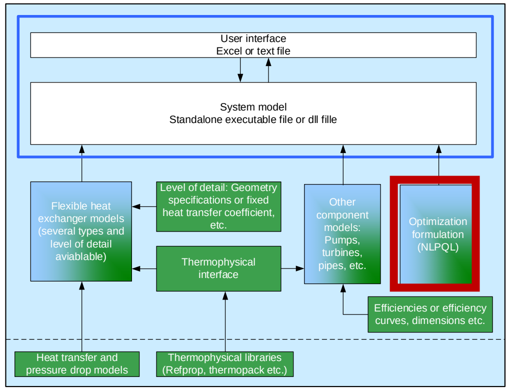


Why rely on in-house models?

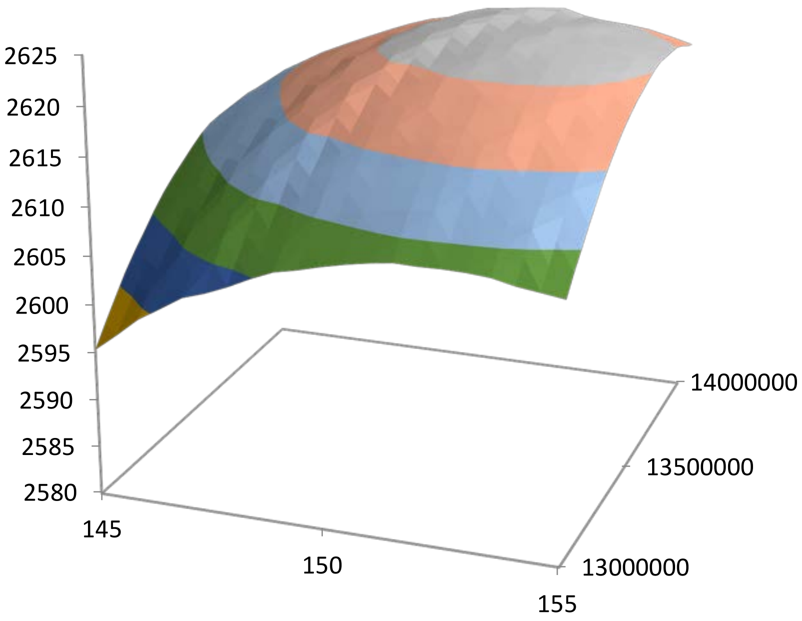
- Required features not available in commercial software
  - Components models
  - System models
  - Off-design behaviour
  - Optimization
- Tailored to best address the problem
  - Direct implementation of state-of-the-art knowledge (correlations, fluid properties)
  - Models for novel and new components, adding new correlations when required
  - Free choice of optimization object function and constraints
- Full control of problem formulation and solution
  - Equations describing fundamental behaviour
  - Correlations for heat transfer, pressure drop, void fraction
  - Parameterization and discretization

- Our in-house models are built to predict behaviour of fluids, components and systems
- In-house models are used when available commercial software is inadequate

System Solver – NLPQL*



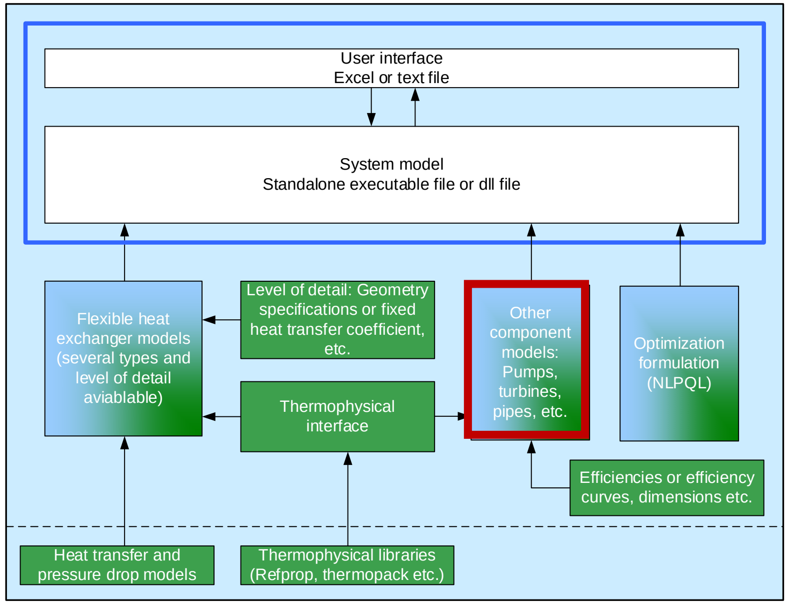
- A constrained non-linear optimisation solver
- Applied in all power cycle models in FlexCS
  - For balancing the system and optimizing the process/HX geometry
- Typical variables: Working fluid mass flow, pressure levels, heat exchanger geometry
- Typical constraints:
  - Avoid two-phase expansion, ensure a closed cycle, maximum allowed heat exchanger area
- Flexible selection of objective function
  - Minimize weight, maximize net power etc.

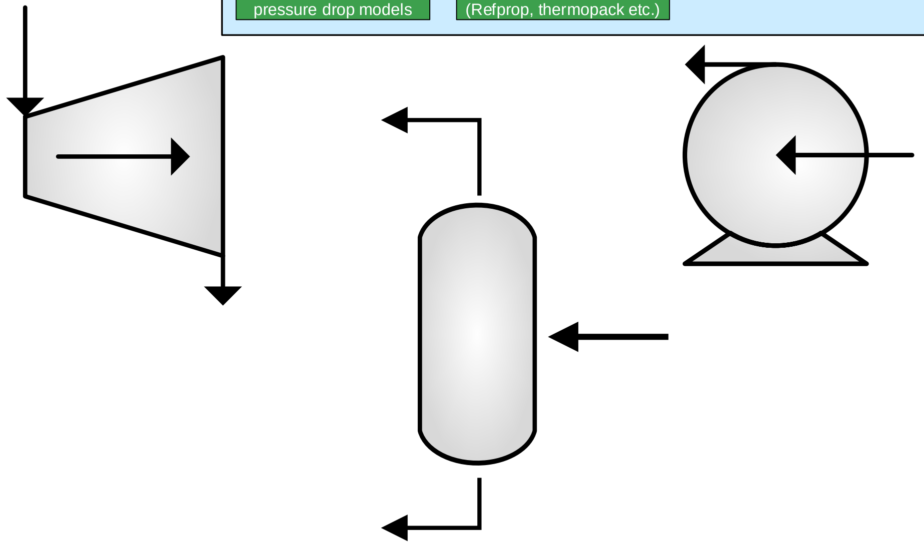


** Schittkowski K (1986) NLPQL: A Fortran subroutine for solving constrained nonlinear programming problems. Annals of Operations Research 5(2):485–500.*

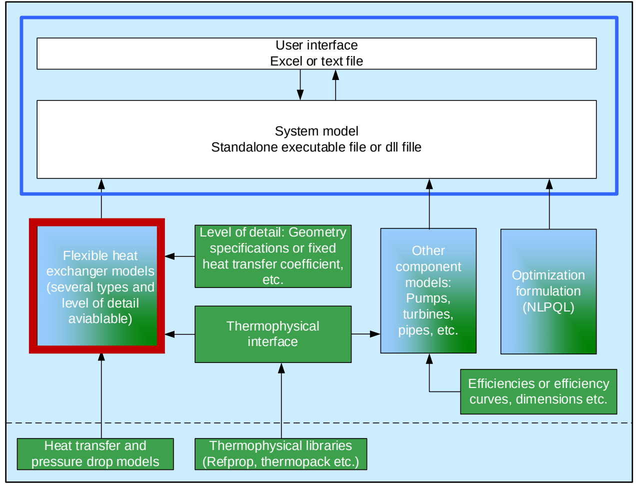
Component models (in addition to heat exchangers)

- Compressors / Pump
  - Isentropic and isochoric
- Expanders / Turbine (Three alternatives)
  - Fixed isentropic efficiency
  - Efficiency curve for a radial expander with Variable Inlet Guide Vanes (VIGV)
  - 1D radial turbine model under development
- Valve
  - Isenthalpic
- Separators
  - Liquid receiver, Suction drum, Flash tank
- Piping
  - FlexHX tube model
- Mixers and splitters
  - Adiabatic



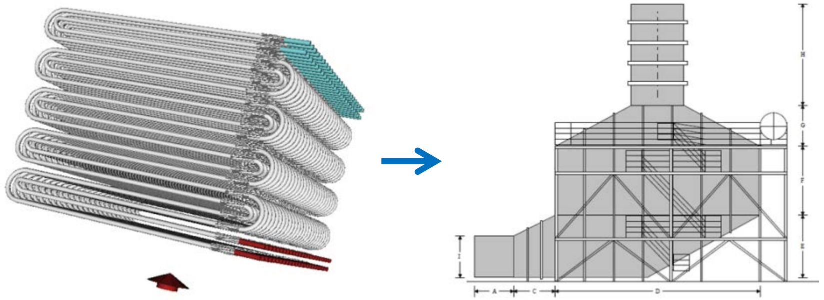


Flexible heat exchanger model selection



- Simple heat exchanger models
  - Pinch point or UA-based models
- Simplified geometry based models
  - Independent of heat exchanger type
  - Characterize the heat exchanger by:
    - Tube diameter, length and number of tubes
  - Obtain estimates for local heat transfer coefficients and pressure drop and total heat transfer area
- FlexHX – flexible heat exchanger library*
  - Detailed models based on geometry and local fluid behaviour.
  - Heat exchanger types:
    - Printed circuit, Plate type, Shell and tube, Tube-in-fin, Finned tube, …

A detailed model of a finned tube waste heat recovery unit



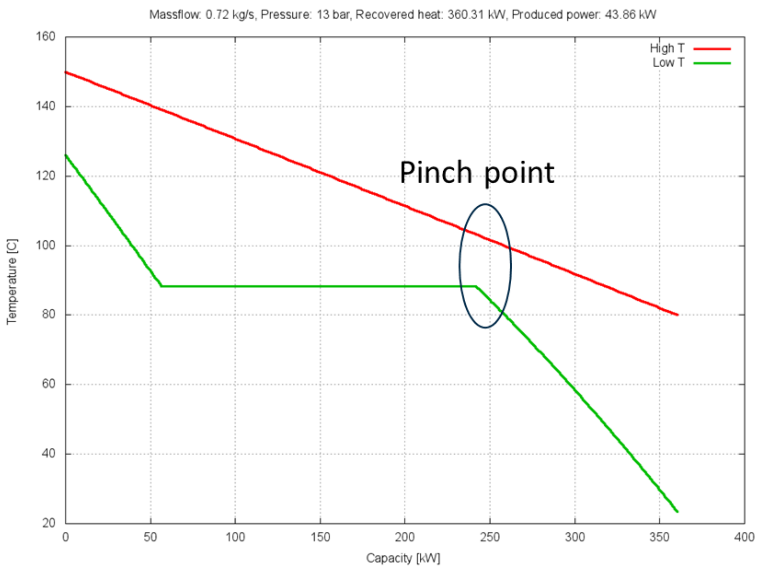
Skaugen Geir, et. al (2014). DESIGN AND OPTIMIZATION OF WASTE HEAT RECOVERY UNIT USING CARBON DIOXIDE AS COOLING FLUID. ASME Power 2014 Conference

* *Skaugen G., Kolsaker K., Walnum H. T., Wilhelmsen Ø. (2013)* A flexible and robust modelling framework for multi-stream heat exchangers. *Computers and Chemical Engineering 49, 95–104*.

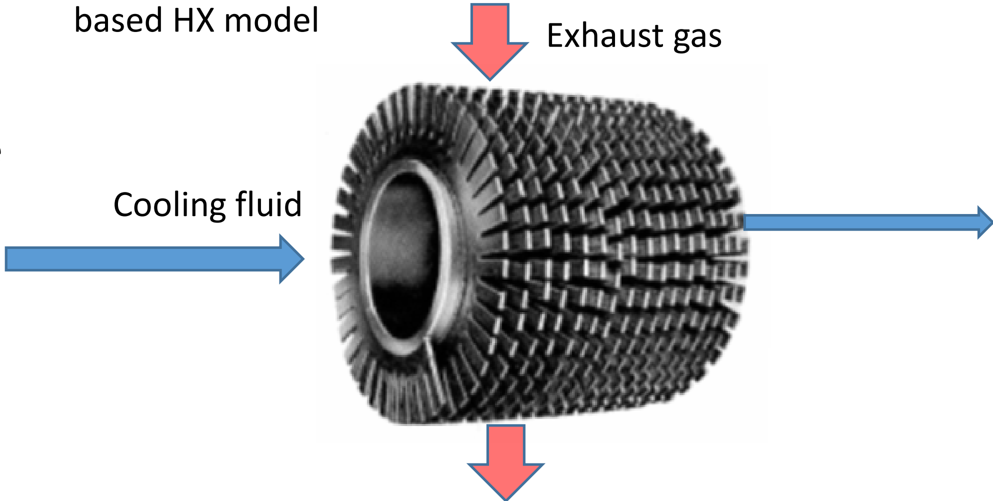
CENTRE FOR ENVIRONMENT-FRIENDLY ENERGY RESEARCH
The Research Council of Norway

Different level of detail of heat exchanger models for different tasks

- Pinch-point or UA-based models
  - Estimate power production potential from a given heat source
  - Screening of heat to power technologies for a given application
  - Working fluid screening
- Simplified geometry models
  - Working fluid screening with few candidate working fluids
    - More fair comparison than UA-based models as pressure drop and heat transfer coefficients are based on geometry and fluid properties
  - Initial cycle evaluation of a more detailed study
  - Initial off-design evaluation
    - Especially when HX-type is not determined
- Detailed geometry based models
  - Detailed case studies taking into account heat exchanger type and size
  - Off-design evaluation
  - Component sizing and optimization



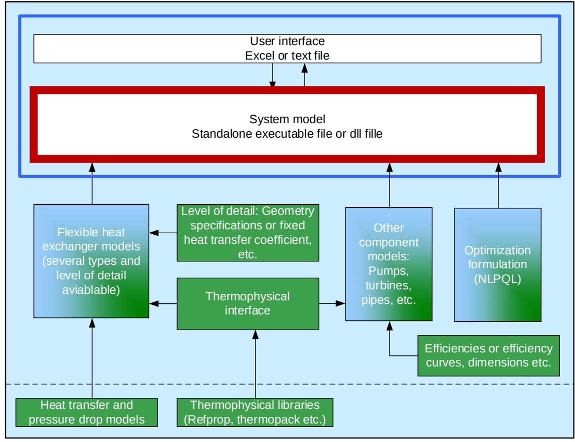
A typical level of detail for geometry based HX model



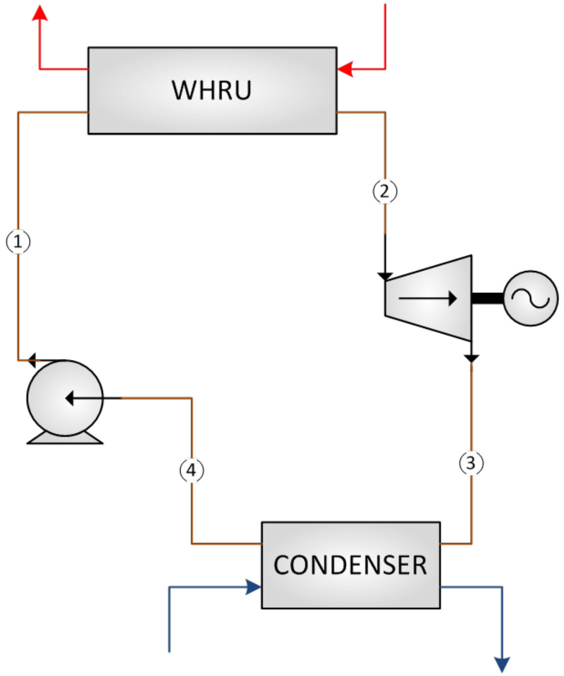
HighEFF

FME CENTRE FOR ENVIRONMENT-FRIENDLY ENERGY RESEARCH
The Research Council of Norway

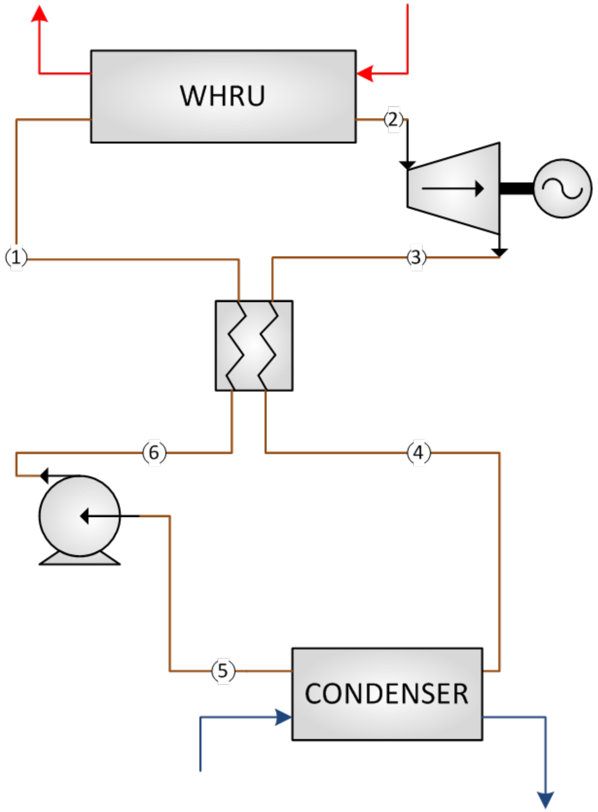
FlexCS – Available models



- Single stage direct heat recovery systems
  - With or without internal heat recovery
  - Several versions exists in terms of level of details
    - Pinch point analysis
    - UA-based analysis
    - Geometry based FlexHX models
    - Expander efficiency curves

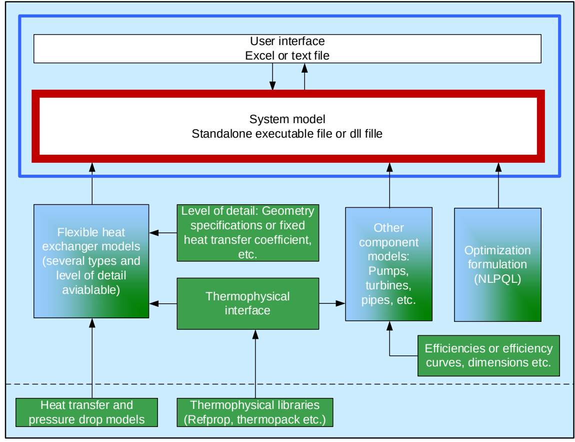


Non-recuperated



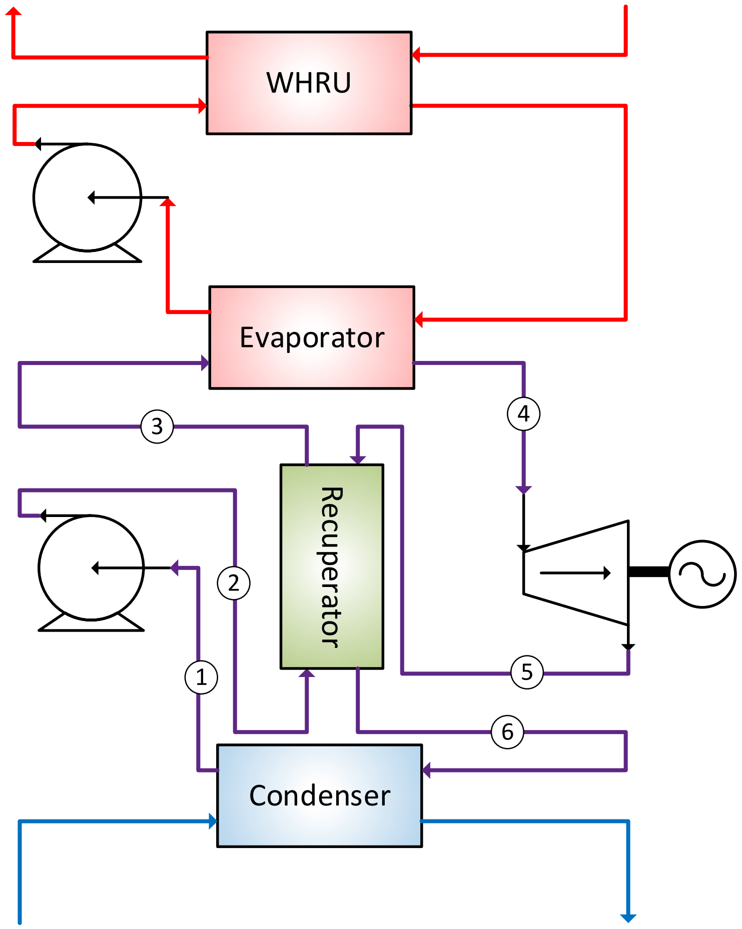
Recuperated

HighEFF



CENTRE FOR ENVIRONMENT-FRIENDLY ENERGY RESEARCH
The Research Council of Norway

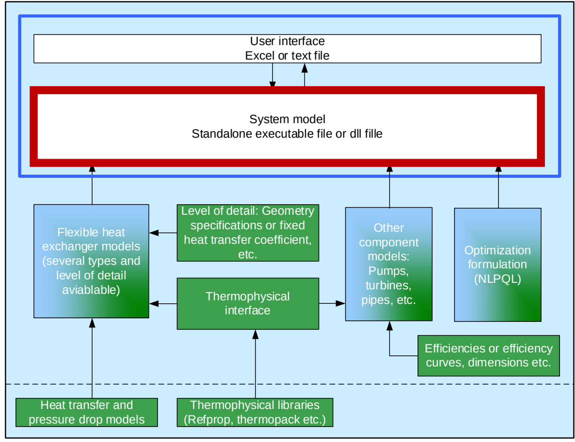
FlexCS – Available models



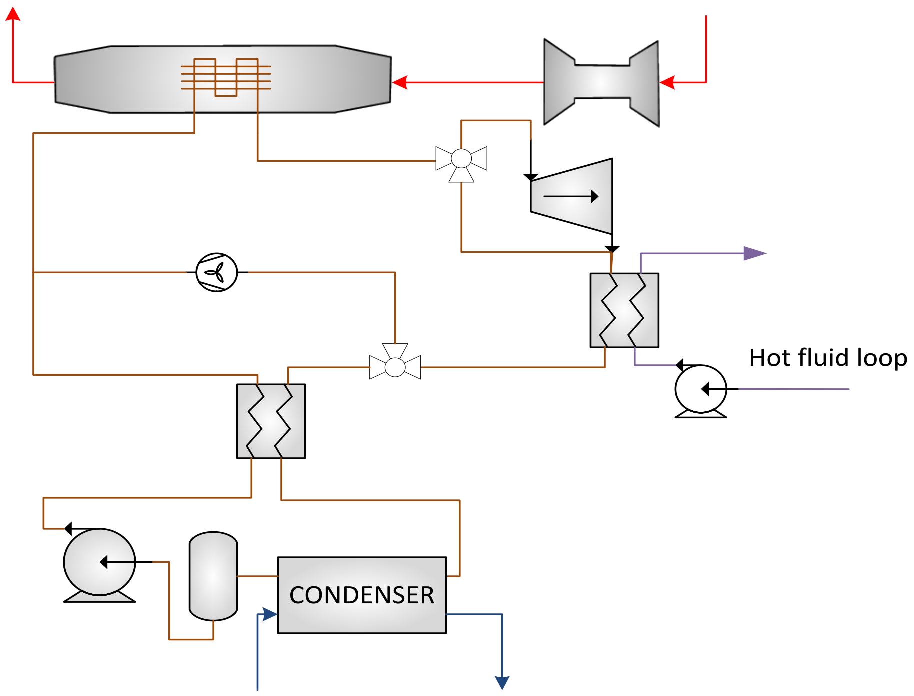
- Single stage indirect heat recovery systems
  - With or without internal heat recovery
  - Developed in HighEFF WP3.1 in 2017
  - Applied by summer researcher Goran Durakovic for analysis of heat recovery from Aluminium industry

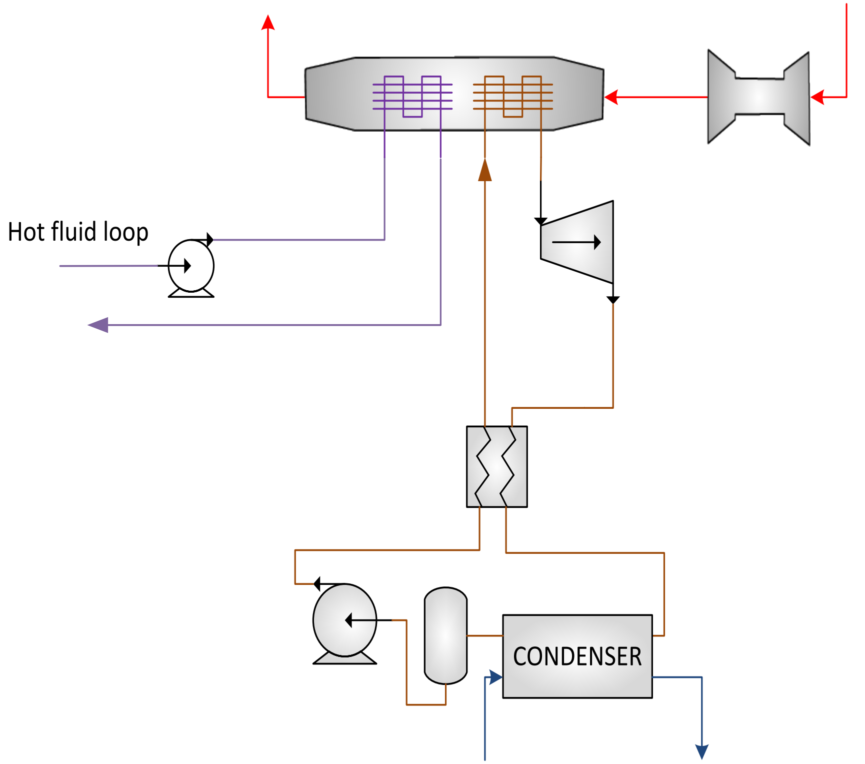
HighEFF

FlexCS – Available models



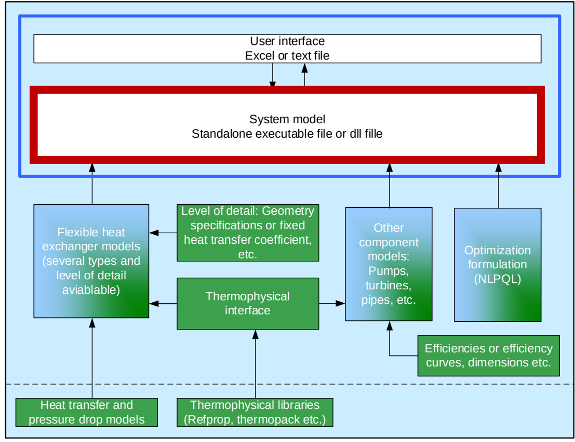
- Combined heat and power cycle
  - Developed in KPN EFFORT
    - Detailed heat exchanger models from FlexHX or
    - UA-based heat exchanger models

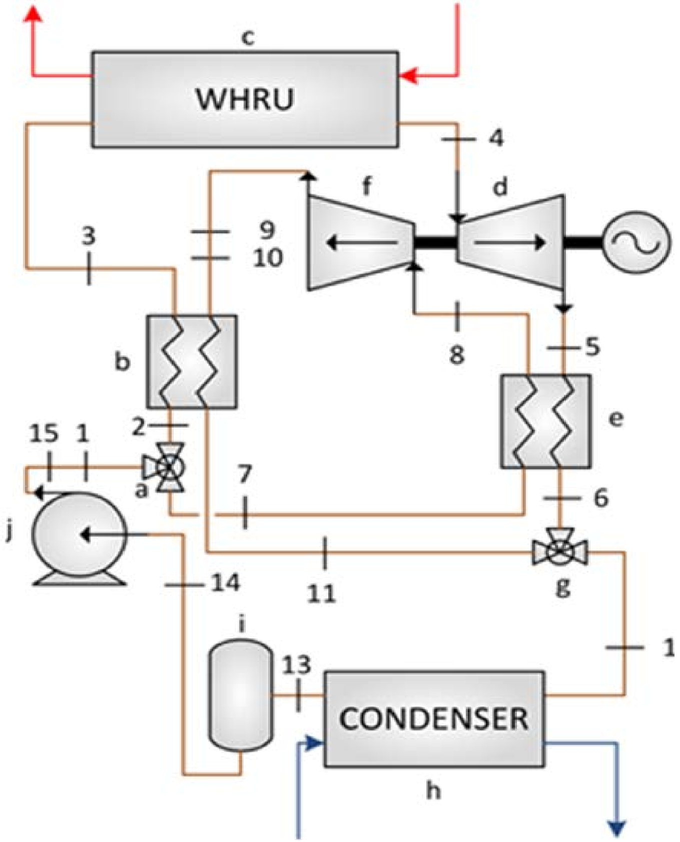




CENTRE FOR ENVIRONMENT-FRIENDLY ENERGY RESEARCH
The Research Council of Norway

FlexCS – Available models





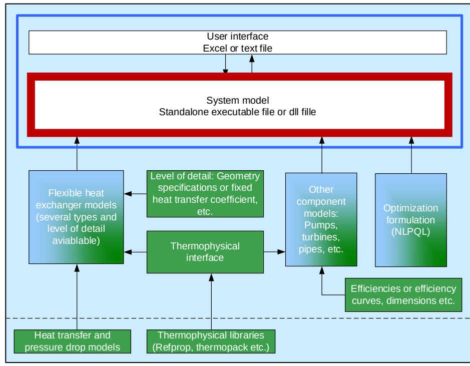
- Dual stage heat recovery systems*
  - Developed in KPN EFFORT
  - A bottoming cycle model for CO_2 as working fluid
  - Detailed heat exchanger models from FlexHX
  - Also applied in industry projects investigating off-design conditions and part load

HighEFF

* *Walnum H. T., Nekså P., Andresen T. (2013)* Modelling and simulation of CO2 (carbon dioxide) bottoming cycles for offshore oil and gas installations at design and off-design conditions. *Computers and Chemical Energy 59, 513–520*.

CENTRE FOR ENVIRONMENT-FRIENDLY ENERGY RESEARCH
The Research Council of Norway

Summary of available models



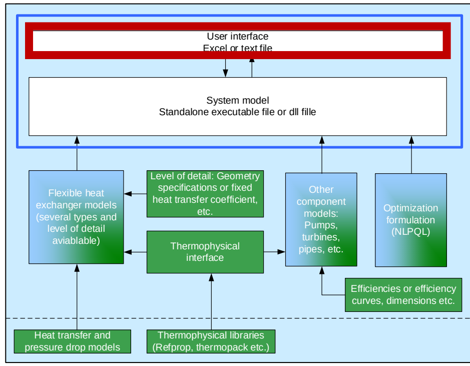
More advanced heat exchanger models towards right

	Pinch analysis	UA-based analysis	Simplified geometry	FlexHX HX models
Single stage direct heat recovery	YES*	YES*	YES*	YES
Single stage indirect heat recovery	YES*	YES	NO	NO
Combined heat and power cycle	NO	YES	NO	YES
Dual stage/ Multiple pressure level	NO	NO	NO	YES (CO_2 bottoming cycle)

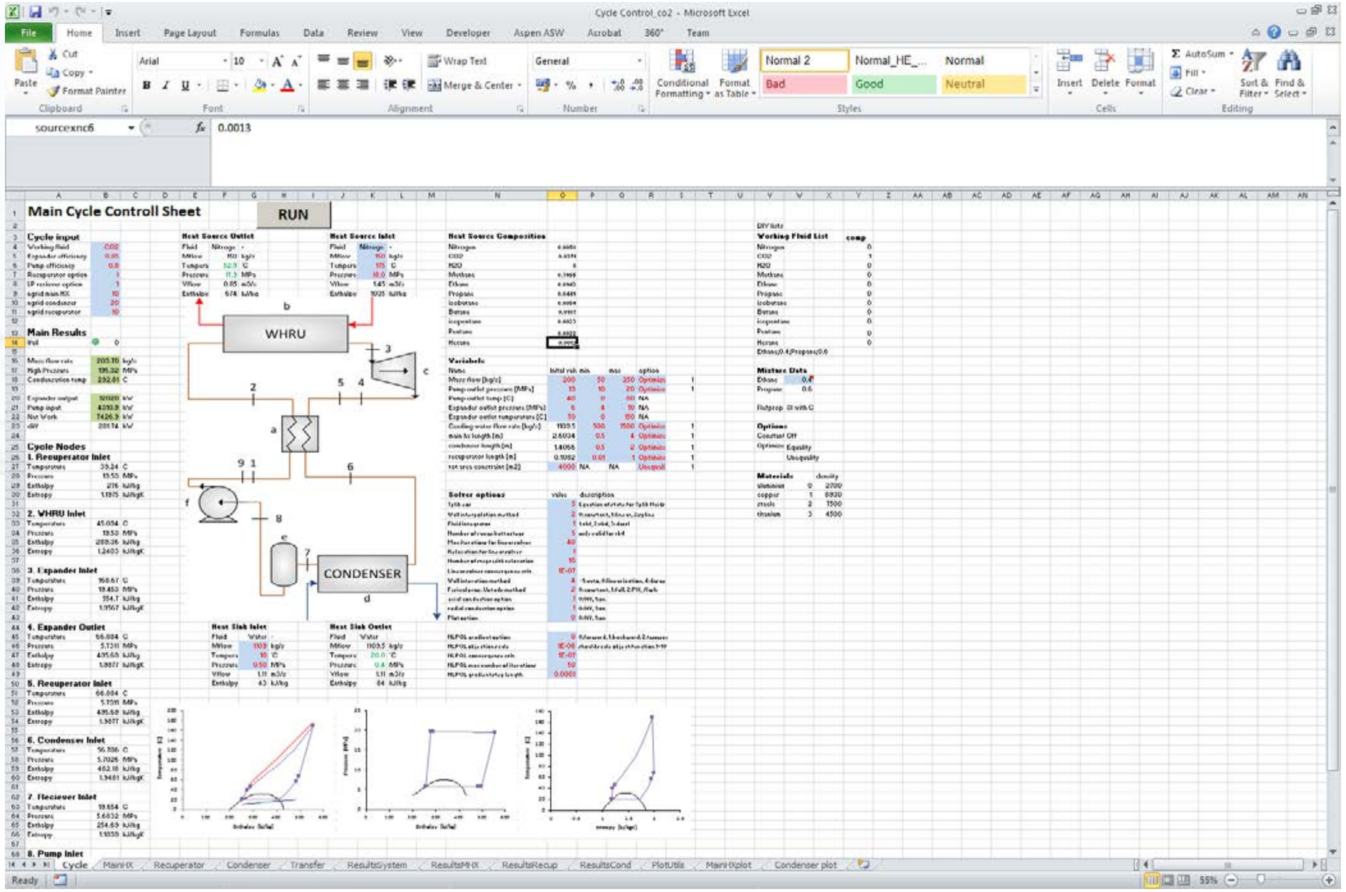
* Recently developed

HighEFF

User interface



Excel



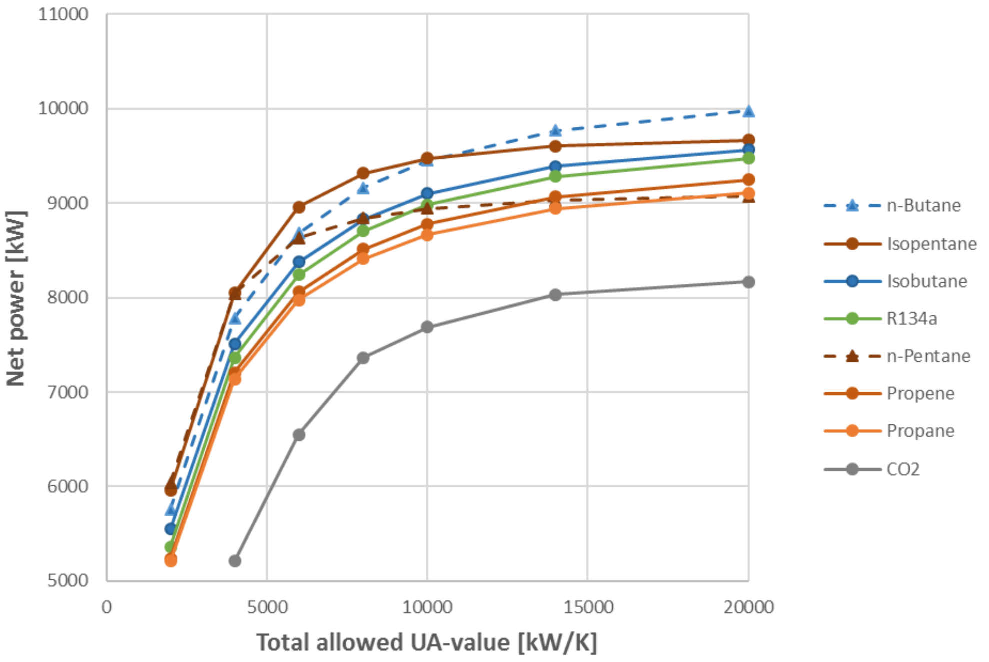
Text file

```
# OPTIMIZATION VARIABLES:

#-varmfl_sink -mfl_sink_min:5.0 -mfl_sink_max:20.0 Fixed

-p_wf_highset:55.0E5 # Fixed high pressure if not variable
-varp_wf_high -p_wf_highmin:50.0E5 -p_wf_highmax:60.0E5

-mfl_wf_set:0.8 # Fixed mass flow if not variable
-varmfl_wf -mfl_wf_min:0.5 -mfl_wf_max:3.5

-T_wf_highset:130
-varT_wf_high -T_wf_highmin:100 -T_wf_highmax:150

# Define low pressure directly or inderictly by condensation temperature
#-p_wf_lowset:10e5
#-varp_wf_low -p_wf_lowmin:3e5 -p_wf_lowmax:14e5

-T_wf_condset:23.5
-varT_wf_cond -T_wf_condmin:13 -T_wf_condmax:40

# CONTRAINTS
-evap_min_DT:2
-cond_min_DT:2
-minSourceTemperature:80
-max_total_A_simple_hx:51.2927
#-fixed_subcooling:0
-expInletSuperheat:
-expOutletSuperheat:


########  NLPQL (solver) parameters
-nlmaxit:30
-nlobjscale:1.0E-4 # Scale factor for objective function
-nlgradopt:2       # Method for numerical differentiation. 0: Forward, 1: Bacwka
rd, 2: cantral differentiation
-nleps:1E-6        # Sum of constraint violation ~ sqrt(nleps)
```

FlexCS – Recent development

- Developed simplified power cycle models for screening purposes
  - Single stage cycle with pinch point or UA-based HX models
  - Development of a simplified geometry based HX model
    - Tested in the single stage cycle model
  - These models will be applied next year in HIGHEFF WP2.1 for evaluating the most promising heat to power technologies described and discussed in 2017 activities
- Power cycle models of more realistic layout
  - Indirect heat recovery system model
- Fundamental development
  - Added new correlations for heat transfer and pressure drop relevant for natural working fluid mixture

Example of a UA-based analysis for working fluid screening.
Similar analysis can be performed for different heat to power technologies



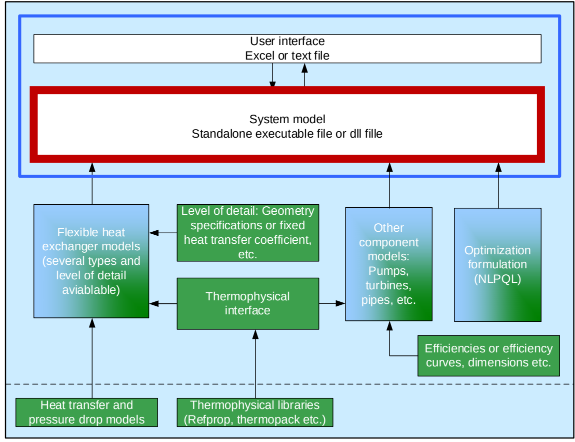
FlexCS – Future work

- Develop simplified models of other heat to power technologies
  - Dual stage/dual pressure systems
  - Absorption cycles (Kalina Cycle for instance)
  - Can be used for a quantitative comparison between heat to power technologies
- Further work on simplified geometry heat exchanger models
  - Currently under development
- Expander model
  - Further development of 1D radial turbine model
  - Addition of efficiency curves for scroll and screw expander
- Develop heat pump models
  - Different technology, but applies similar components

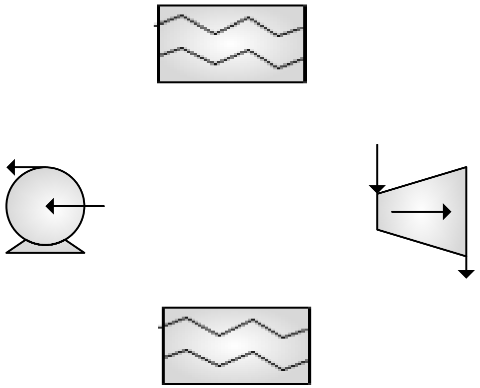
Additional technical slides

- The following slides describe the structure in how to build and solve a system model

Building and solving system model – The API



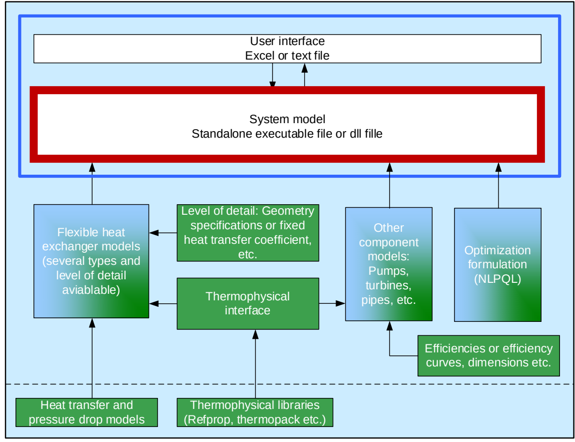
- Add components
  - Describing geometry and performance parameters



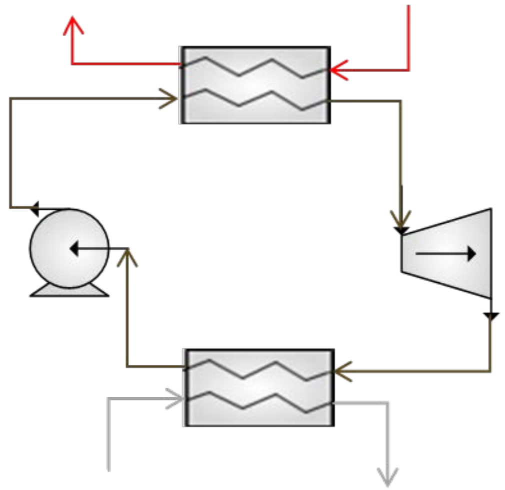
```
FCSHx_createFixedUA("evaporator");
FCSSystem_registerHx(system,evaporator);
FCSExpander_createIsentropic("expander", expInlet, expOutlet, eta, nstages)
FCSSystem_registerExpander(system, exp);
…
```

HighEFF

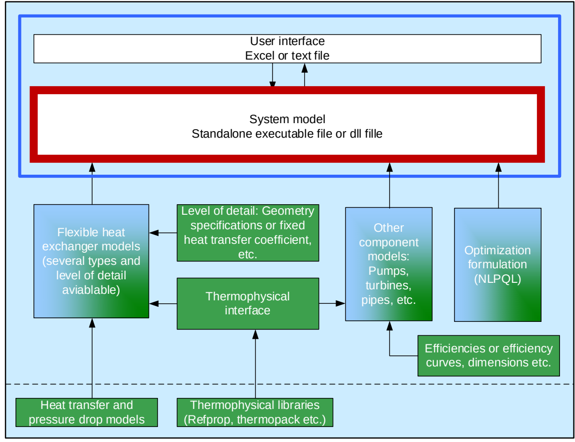
Building and solving system model – The API



- Add components
  - Describing geometry and performance parameters
- Add the streams
  - Contains fluid properties (p, h, $\dot{m}$)
  - Connects the components

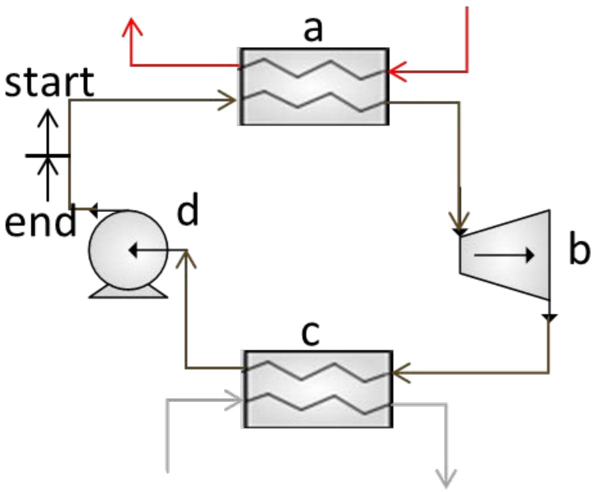


```
evaporatorInlet = FCSStream_create ("evaporatorInlet",wfcompN->buf,
                                    wflibrary, ncompwf, calctransport, mflRef);
FCSStream_setMixture (evaporatorInlet, zREF, PRefHigh, TRefHigh);
FCSStream_setPT (evaporatorInlet, PRefHigh, TRefHigh);
FCSSystem_appendStream (system, evaporatorInlet);
…
```



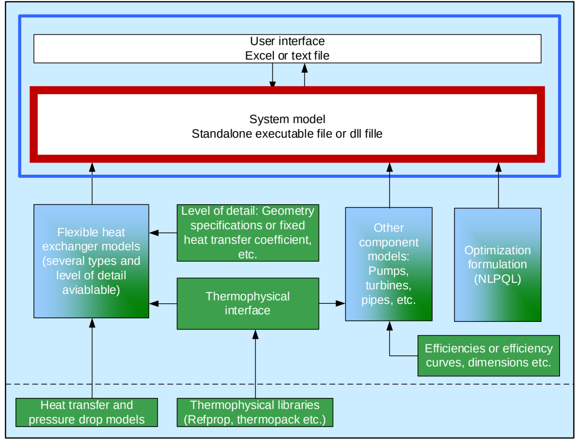
Building and solving system model – The API

- Add components
  - Describing geometry and performance parameters
- Add the streams
  - Contains fluid properties (p, h, $\dot{m}$)
  - Connects the components
- Set calculation sequence
  - Define starting point




```
FCSHx_solveHx(evaporator, savedata);
FCSExpander_CalculateExpander (expander);
FCSHx_solveHx(condenser, savedata);
FCSCompressor_CalculateCompressor (compressor)
…
```

Building and solving system model – The API



- Add components
  - Describing geometry and performance parameters
- Add the streams
  - Contains fluid properties (p, h, $\dot{m}$)
  - Connects the components
- Set calculation sequence
  - Define starting point
- Set solver parameters
  - Iteration streams
  - Variables
  - Equality and inequality constraints
  - Objective function