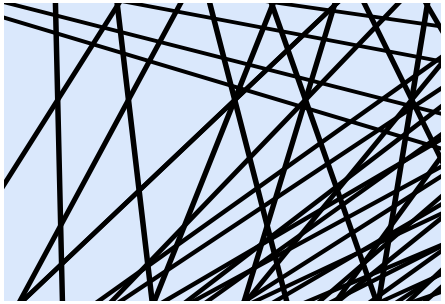
dlbltla

Process Dynamics and Operations Group

BASF We create chemistry

technische universität dortmund

Technische Universität Berlin



Mixed-Integer Optimization in GAMS using CAPE-OPEN Thermodynamics via MOSAICmodeling and COBIA for Conceptual Design

CAPE-OPEN 2019 Annual Meeting | 22.10.2019

David Krone, Erik Esche, Norbert Asprion, Jens-Uwe Repke, Mirko Skiborowski

Motivation
Optimization with thermodynamic calculations

1. Formulate all thermo functions explicitly

$$\begin{aligned} \min \quad & f\,(\mathrm{x}) \\ s.t. \quad & g(x) = 0 \end{aligned}$$

→ Inflated model

- Complex intialization necessary
- No easy changes in thermo calculations (e.g. change from ideal to non-ideal liquid phase)

2. Function calls to thermo property packages

$$\begin{aligned} \min \quad & f\,(\mathrm{x},\mathrm{y}) \\ s.t. \quad & g(x,y) = 0 \\ & y = thermo(x) \end{aligned}$$

→ Lean models with thermo flexibility

Problem: optimization tools do not usually provide integrated thermo packages

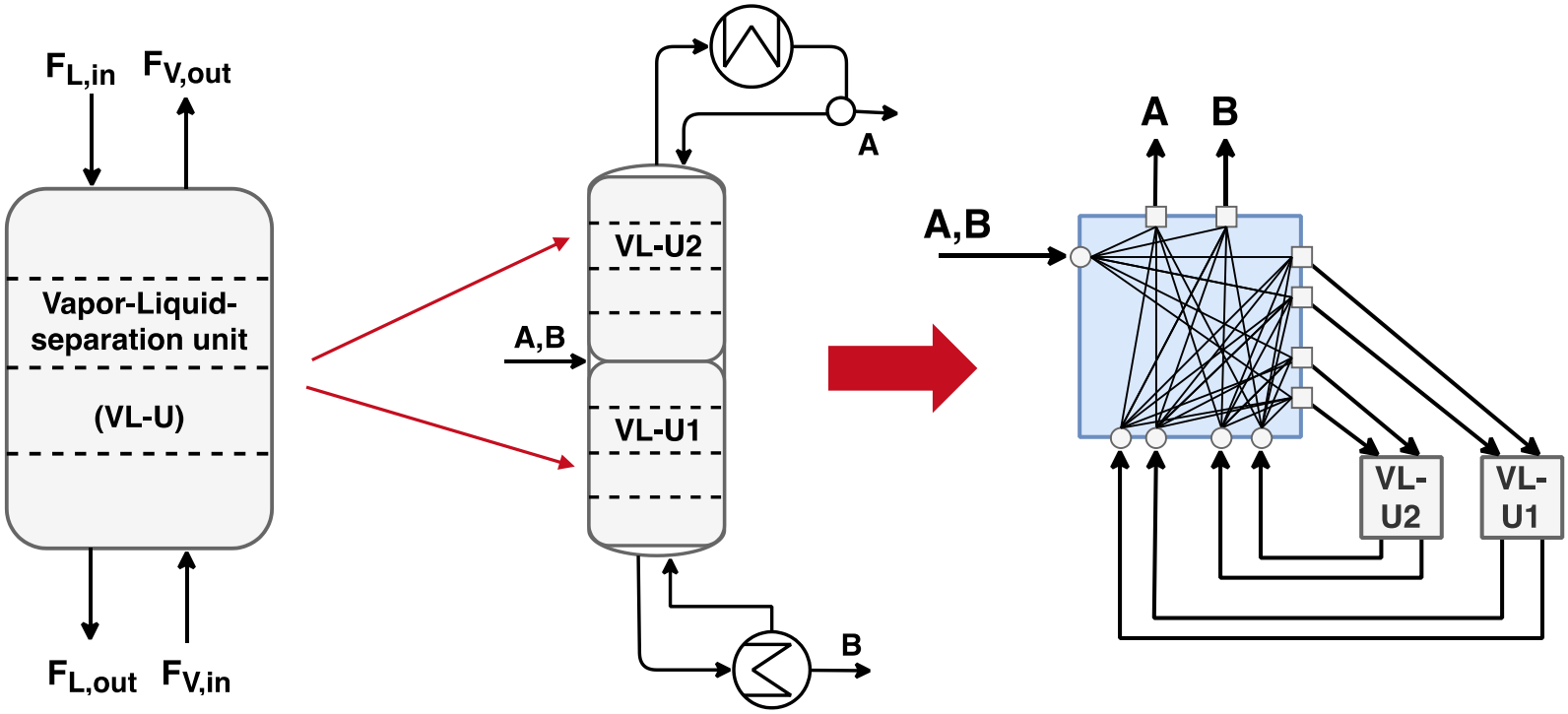
Motivation
Development Goal

Generate the **interface between**

GAMS and CAPE-OPEN thermo property packages

for **fast thermo calculations**

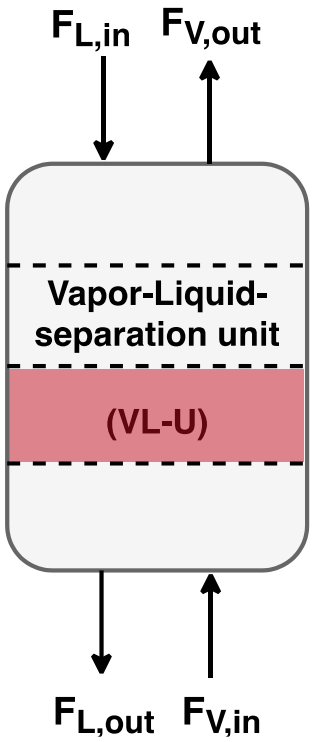
I. Mixed-Integer(MI) Optimization for Conceptual Design



I. MI Optimization for Conceptual Design
The necessity of calculation speed

Number of thermo calls per equilibrium stage (ternary mixture)

property	Calls in GAMS program file	Calls to thermo package (C++)
single phase enthalpies (h^L and h^V)	**2**	**8**
distribution coefficients K_c	**3**	**18**



→ 26 calls/stage

For four equilibrium stages per unit:

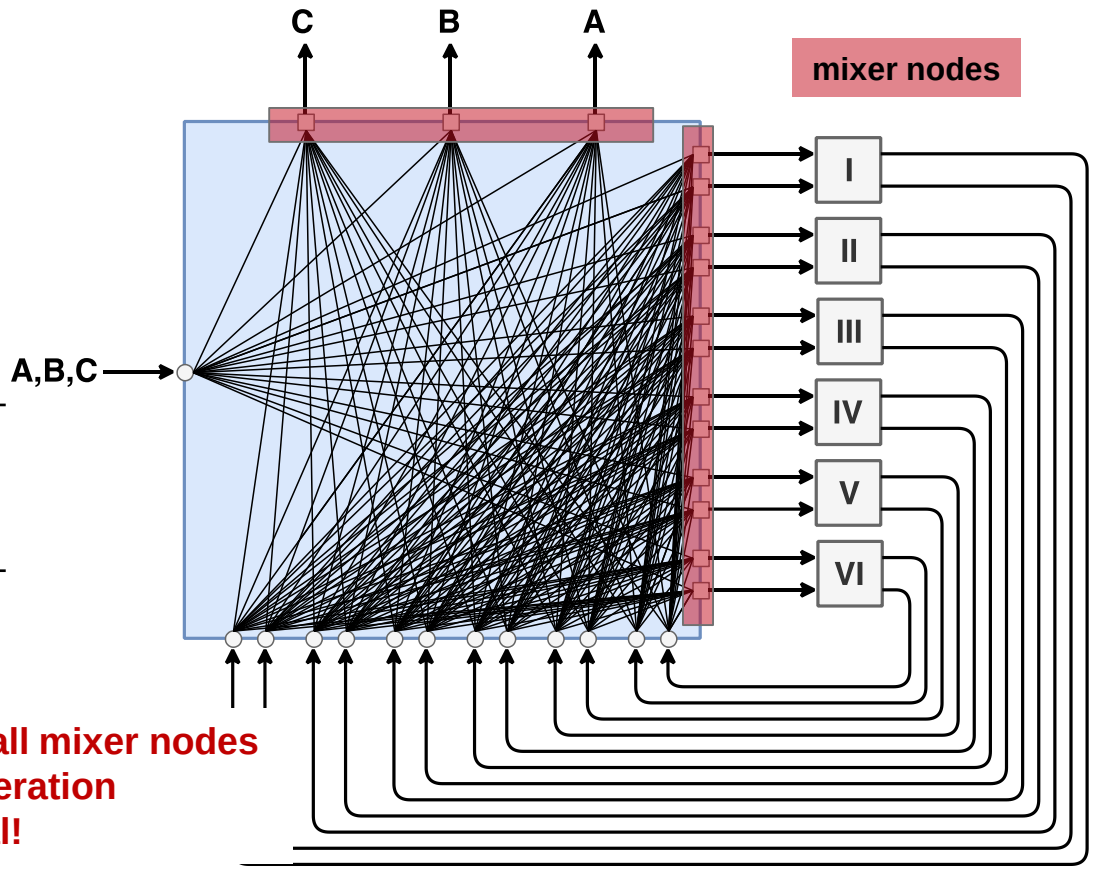
→ 104 calls/separation unit

I. Optimization-based Conceptual Design
The necessity of calculation speed

→ **624 calls/all separation units**

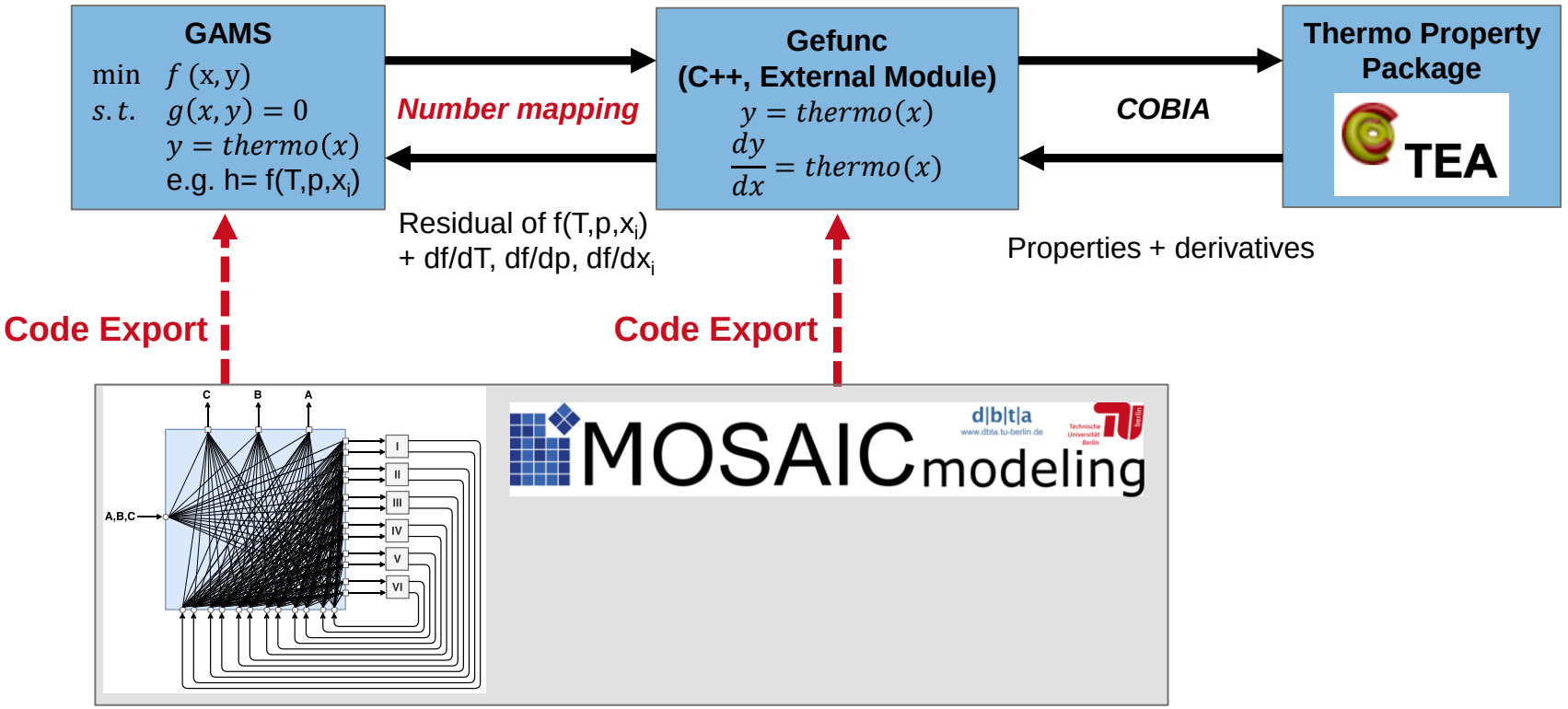
Number of thermo function calls per mixer node

property	Calls to thermo package (C++)
single phase enthalpies (h^L and h^V)	**8**
distribution coefficients K_c	**18**



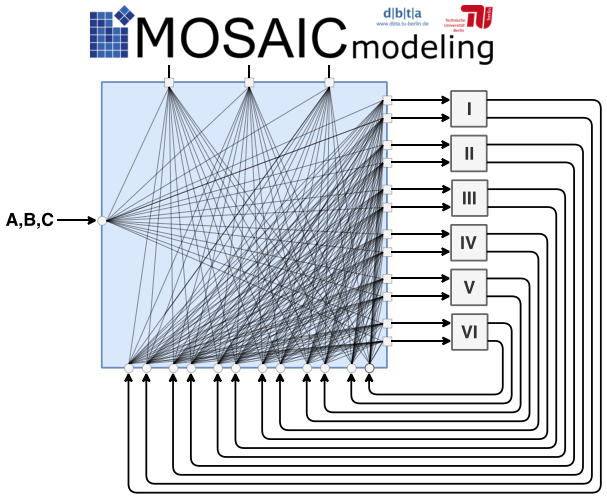
→ **26 calls/mixer node → 390 calls/all mixer nodes**
→ **1014 thermo function calls per iteration**
Calculation speed is vital!

II. Thermo function interface



III. Optimization Workflow Overview

1. Model development



2. Code export

.gms program file

GAMS

Number mapping between external equations (.gms) and external module

External Module
(C++ file → DLL)

GAMS workspace

3. Thermo calls via COBIA

Thermo property package (manager)

TEA

III. Optimization Workflow
1. Model Development – Equation System

Equation System | Notation

Hierarchy | Equations | Functions | Flowsheet

- 124220: EQS_6&VL-PBBs [with cc
  - 0 – [s] 124221: EQS_VL_PBB (
    - 0 – [s] 124222: EQS evap
    - 1 – [s] 124217: EQS conde
    - 2 – [s] 120149: EQS_relax
  - 1 – [s] 124221: EQS_VL_PBB (
  - 2 – [s] 124221: EQS_VL_PBB (
  - 3 – [s] 124221: EQS_VL_PBB (
  - 4 – [s] 124221: EQS_VL_PBB (
  - 5 – [s] 124221: EQS_VL_PBB (
  - 6 – [s] 123506: EQS_DN&flash

All Subsystems of the Superstructure

Connected Elements

(1) $y_{VL,i,j} = \beta_i \cdot K^{VLE}_{VL,i,j} \cdot x_{VL,i,j}$

(2) $F^L_{VL,i-1} \cdot h^L_{VL,i-1} + Q^{trans}_{VL,i} = F^L_{VL,i} \cdot h^L_{VL,i}$

(3) $F^L_{VL,i-1} \cdot x_{VL,i-1,j} + F^{trans}_{VL,i,j} = F^L_{VL,i} \cdot x_{VL,i,j}$

(4) $F^V_{VL,i+1} \cdot h^V_{VL,i+1} - Q^{trans}_{VL,i} = F^V_{VL,i} \cdot h^V_{VL,i}$

(5) $F^V_{VL,i+1} \cdot y_{VL,i+1,j} - F^{trans}_{VL,i,j} = F^V_{VL,i} \cdot y_{VL,i,j}$

(6) $\sum_{i=1}^{nc} x_{VL,i,j} = 1$

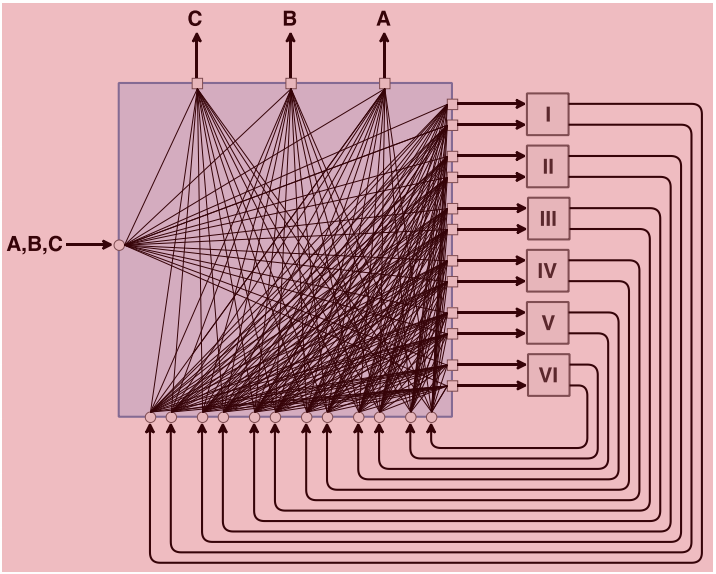
(7) $\sum_{i=1}^{nc} y_{VL,i,j} = 1$

(8) $p^{PBB}_{VL} = p_{VL}$

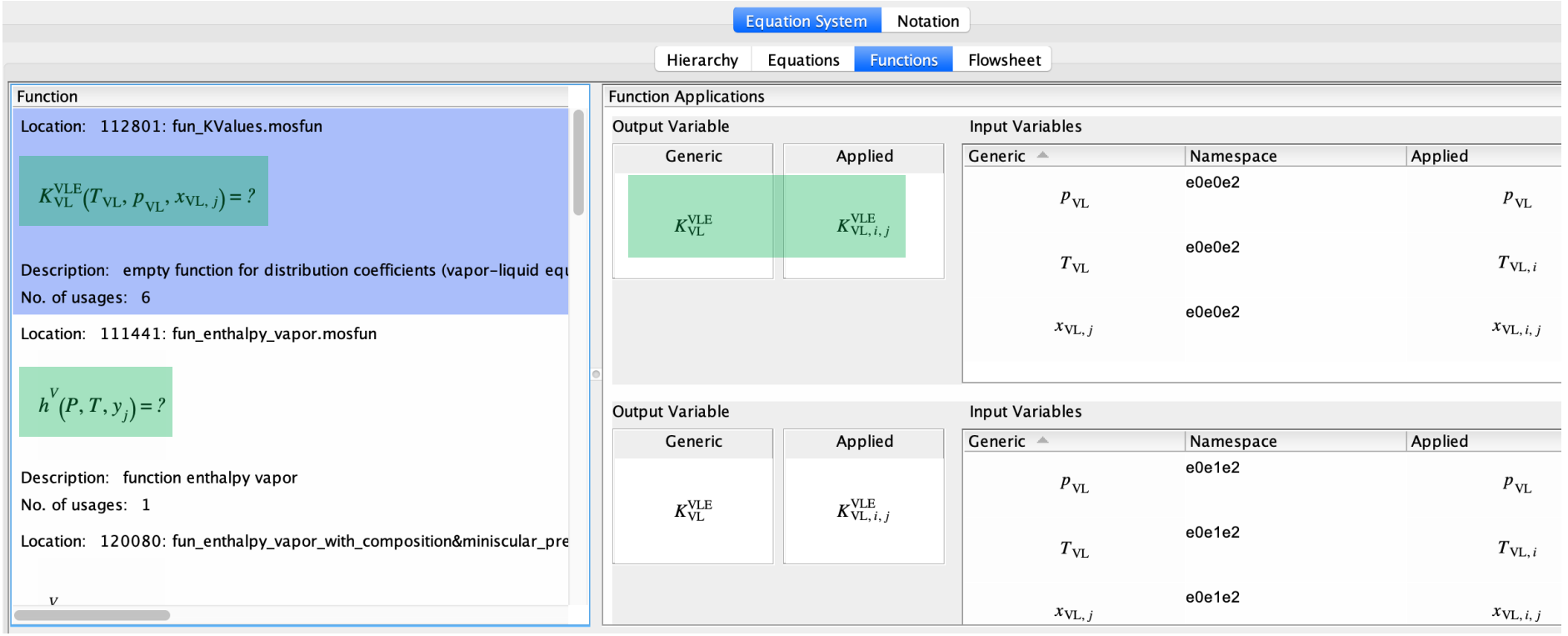
(9) $\beta_i - 1 = s^V_i - s^L_i$

(10) $(s^L_i + t_{tol,i}) \cdot (F^L_{VL,i} + t_{tol,i}) - (t_{tol,i})^{2.0} \geq 0$

System equations
all equations defined generically!
→ instantiated before initialization

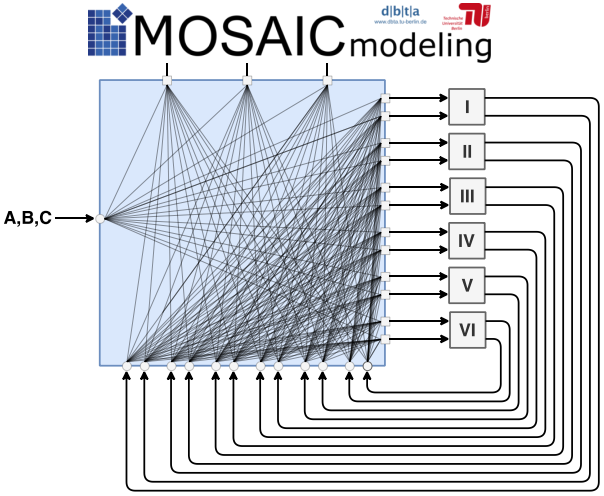


III. Optimization Workflow
1. Model Development – Thermo Functions



III. Optimization Workflow
2. Code Export

1. Model development ✓



2. Code export

.gms program file

GAMS

Number mapping between external equations (.gms) and external module

External Module
(C++ file → DLL)

GAMS workspace

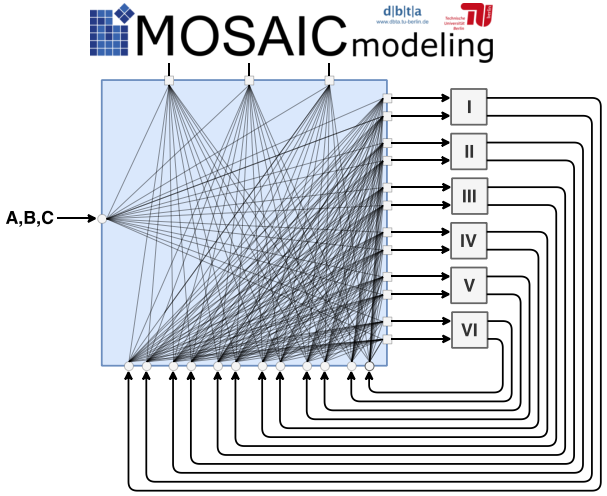
3. Thermo calls via COBIA

Thermo property package (manager)

TEA

III. Optimization Workflow
2. Code Export

1. Model development ✓



2. Code export ✓

.gms program file GAMS

Number mapping between external equations (.gms) and external module

External Module (C++ file → DLL)

GAMS workspace

3. Thermo calls via COBIA

Thermo property package (manager)

TEA

III. Optimization Workflow
Mapping of external equations and external module

```
=x= 1*e0e6_h_outL_mix_k1 + 43*e0e6_x_outL_mix_k1_j1 + 44*e0e6_x_outL_mix_k1_j2 + 45*e0e6_x_outL_mix_k1_j3 + 31*e0s0_T + 37*e0s0_p;
=x= 2*e0e6_h_outL_mix_k2 + 46*e0e6_x_outL_mix_k2_j1 + 47*e0e6_x_outL_mix_k2_j2 + 48*e0e6_x_outL_mix_k2_j3 + 32*e0s1_T + 38*e0s1_p;
=x= 3*e0e6_h_outL_mix_k3 + 49*e0e6_x_outL_mix_k3_j1 + 50*e0e6_x_outL_mix_k3_j2 + 51*e0e6_x_outL_mix_k3_j3 + 33*e0s2_T + 39*e0s2_p;
=x= 4*e0e6_h_outL_mix_k4 + 52*e0e6_x_outL_mix_k4_j1 + 53*e0e6_x_outL_mix_k4_j2 + 5[illegible]e0e6_x_outL_mix_k4_j3 + 34*e0s3_T + 40*e0s3_p;
=x= 5*e0e6_h_outL_mix_k5 + 55*e0e6_x_outL_mix_k5_j1 + 56*e0e6_x_outL_mix_k5_j2 + 5[illegible]0e6_x_outL_mix_k5_j3 + 3[illegible];
=x= 6*e0e6_h_outL_mix_k6 + 58*e0e6_x_outL_mix_k6_j1 + 59*e0e6_x_outL_mix_k6_j2 + 60*e0e6_x_outL_mix_k6_j3 + 3[illegible];
=x= 7*e0e6_h_outV_mix_k1 + 61*e0e6_x_outV_mix_k1_j1 + 62*e0e6_x_outV_mix_k1_j2 + 63*e0e6_x_outV_mix_k1_j3 + 3[illegible];
=x= 8*e0e6_h_outV_mix_k2 + 64*e0e6_x_outV_mix_k2_j1 + 65*e0e6_x_outV_mix_k2_j2 + 66*e0e6_x_outV_mix_k2_j3 + 32*e0s1_T + 38*e0s1_p;
```

.gms program file

GAMS

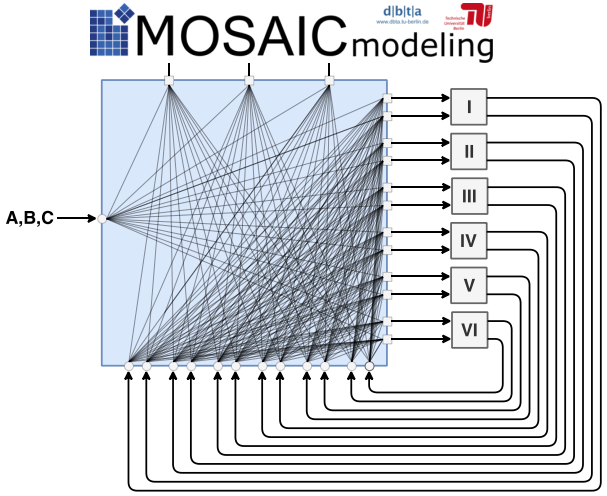
Referencing by numbers, e.g. for: $h^L = f(p, T, x_1, x_2, x_3)$

```
if (icntr[I_Eqno] == 1) {//for the first function
    std::vector<double> composition_liquid = { x[42], x[43], x[44] };
    //External Function for liquid enthalpy calculation
    mylocalThermoObj.setSinglePhasePropResult(enthalpyF, x[30], x[36] * 100000.0, liquid, composition_liquid);
    f[0] = (x[0] - mylocalThermoObj.getSinglePhasePropResult(0));
    //Derivatives
    //dh/dh
    d[0] = 1.0;
    //dh/dy_j
    mylocalThermoObj.setSinglePhasePropResult(enthalpyF_DmolFraction, x[30], x[36] * 100000.0, liquid, composition_liquid);
    d[42] = -mylocalThermoObj.getSinglePhasePropResult(0);
    d[43] = -mylocalThermoObj.getSinglePhasePropResult(1);
    d[44] = -mylocalThermoObj.getSinglePhasePropResult(2);
    //dh/dp
    mylocalThermoObj.setSinglePhasePropResult(enthalpyF_Dpressure, x[30], x[36] * 100000.0, liquid, composition_liquid);
    d[36] = -mylocalThermoObj.getSinglePhasePropResult(0)*100000.0;//[1/bar]*/
    //dh/dT
    mylocalThermoObj.setSinglePhasePropResult(enthalpyF_Dtemperature, x[30], x[36] * 100000.0, liquid, composition_liquid);
    d[30] = -mylocalThermoObj.getSinglePhasePropResult(0);//[1/K]
}
```

External Module
(C++ file → DLL)

III. Optimization Workflow
3. Thermo calls to Thermo package via COBIA

1. Model development ✓



2. Code export ✓

.gms program file

GAMS

Number mapping between external equations (.gms) and external module

External Module
(C++ file → DLL)

GAMS workspace

3. Thermo calls via COBIA ✓

Thermo property package (manager)

TEA

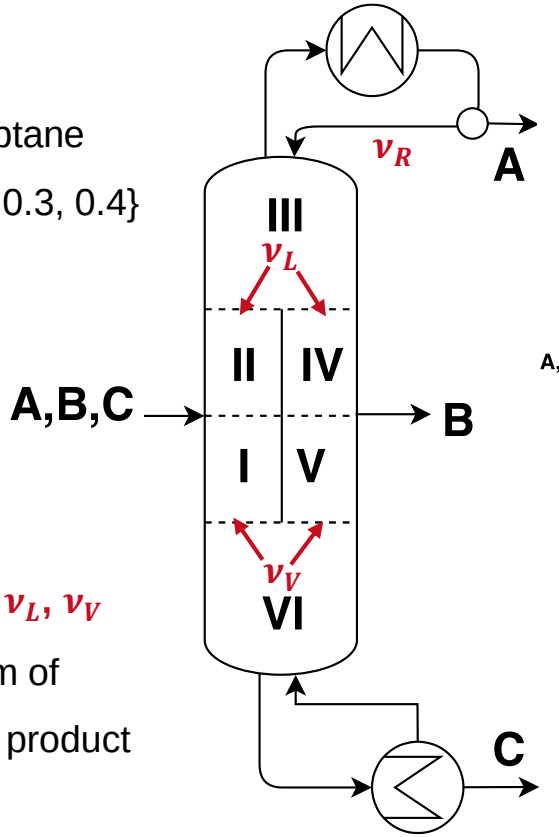
IV. Case Studies – Separation of ternary mixture via a dividing wall column (DWC)

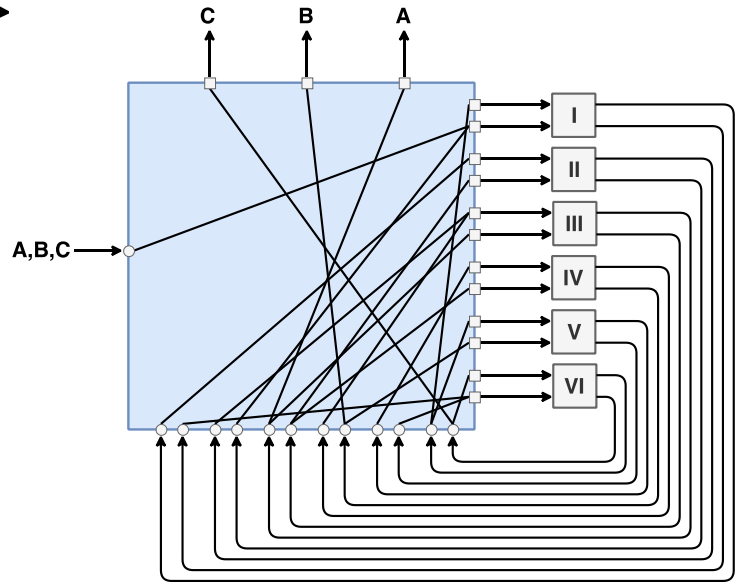
- **Feed specifications:**
  - n-pentane/n-hexane/n-heptane
  - Molar composition = {0.3, 0.3, 0.4}
  - T = 325 K
  - p = 1 bar
  - F = 1 kmol/s

- **Optimization task:**

  find optimal split ratios: ν_R, ν_L, ν_V

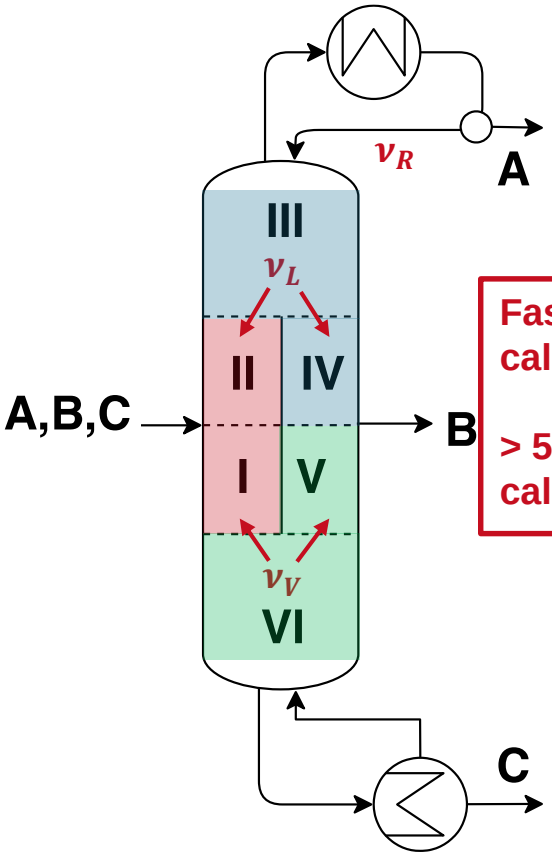
- **Maximizing objective:** sum of desired product qualities in product streams





Only showing active superstructure connections

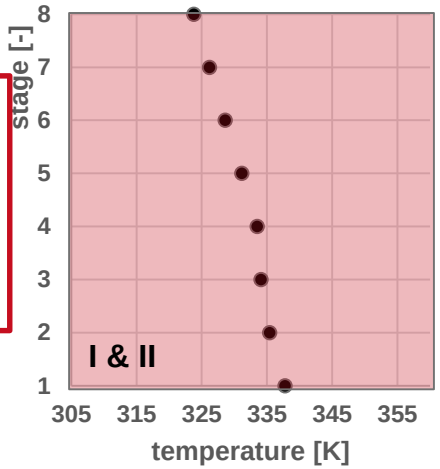
IV. Case Studies – Separation of ternary mixture
Speed of thermo calculations and profiles

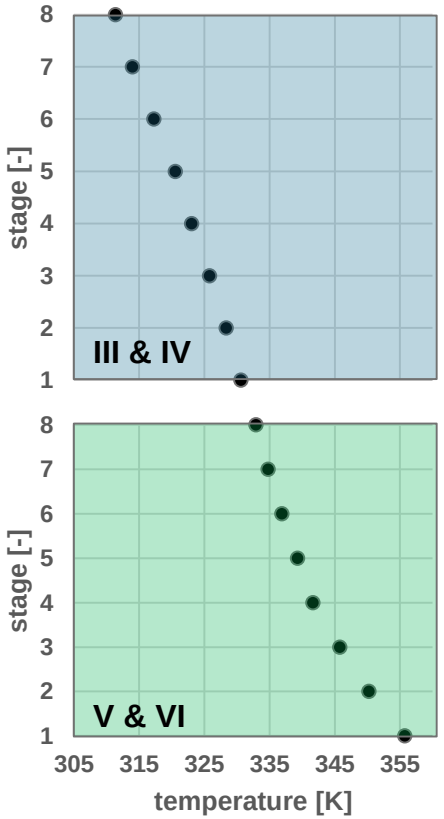


Fast thermo calculations

> 500 thermo calculations/second

$\nu_R = 0.4$
$\nu_L = 0.69,$
$\nu_V = 0.45$





V. Summary

Functioning interface between

GAMS and CAPE-OPEN thermo property packages

For fast thermo calculations

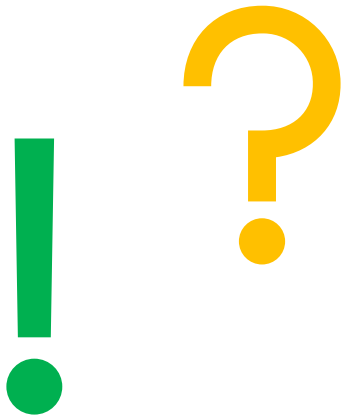
Reducing modelling effort by automatic code generation

with MOSAICmodeling

Acknowledgements

Jasper van Baten
for his code examples and help with COBIA

Questions and answers



II. Results – separation of ternary mixture
ii. Optimization of the superstructure in DWC configuration

Product flows and qualities:

Destillate:

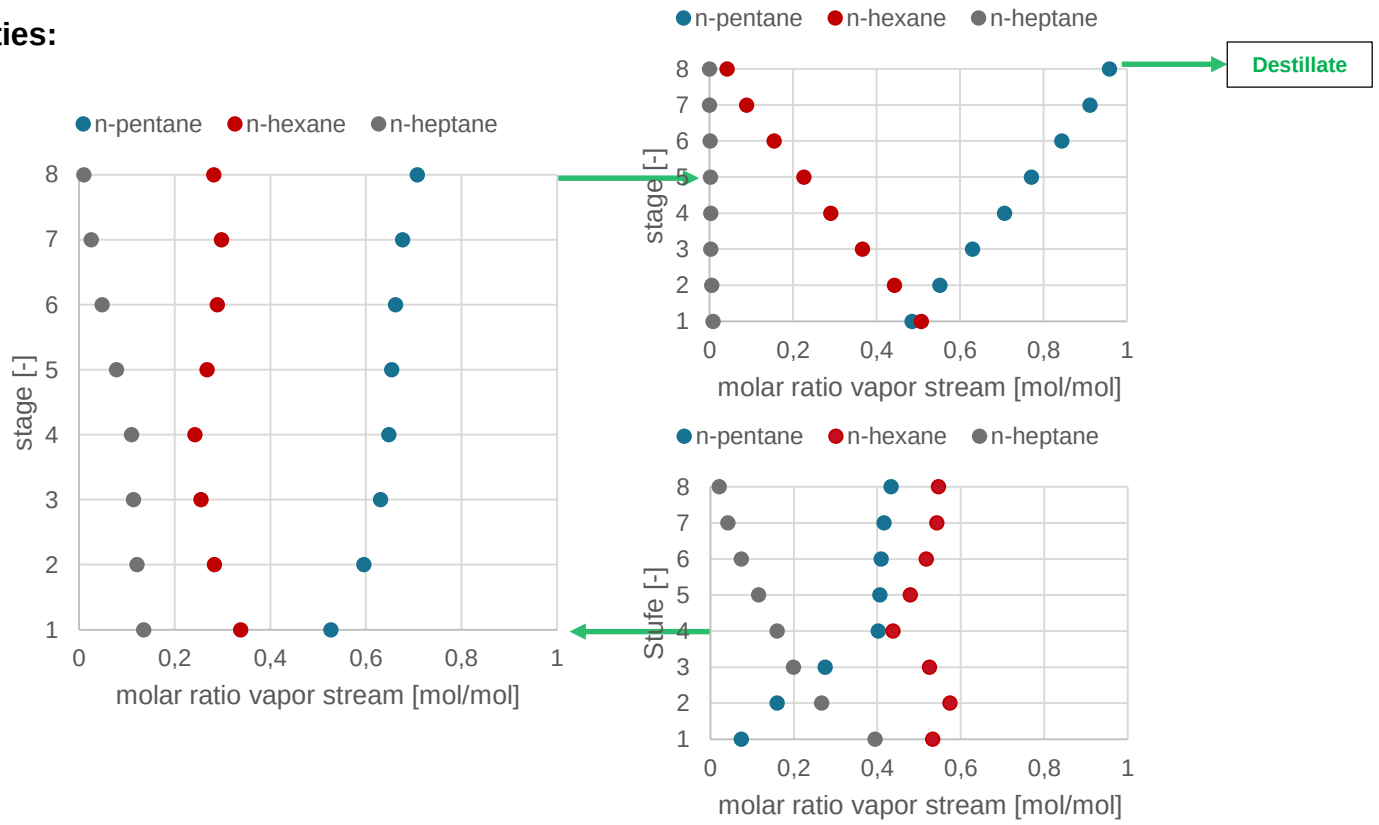
- F = 0,2752 kmol/s
- Composition = {**0.958**, 0.042, 0.000}

Middle draw:

- F = 0,1 kmol/s
- Composition = {0.241, **0.726**, 0.032}

Bottom draw:

- F = 0,6236 kmol/s
- Composition = {0.019, 0.345, **0.636**}



II. Results – separation of ternary mixture
ii. Optimization of the superstructure in DWC configuration

Product flows and qualities:

Destillate:

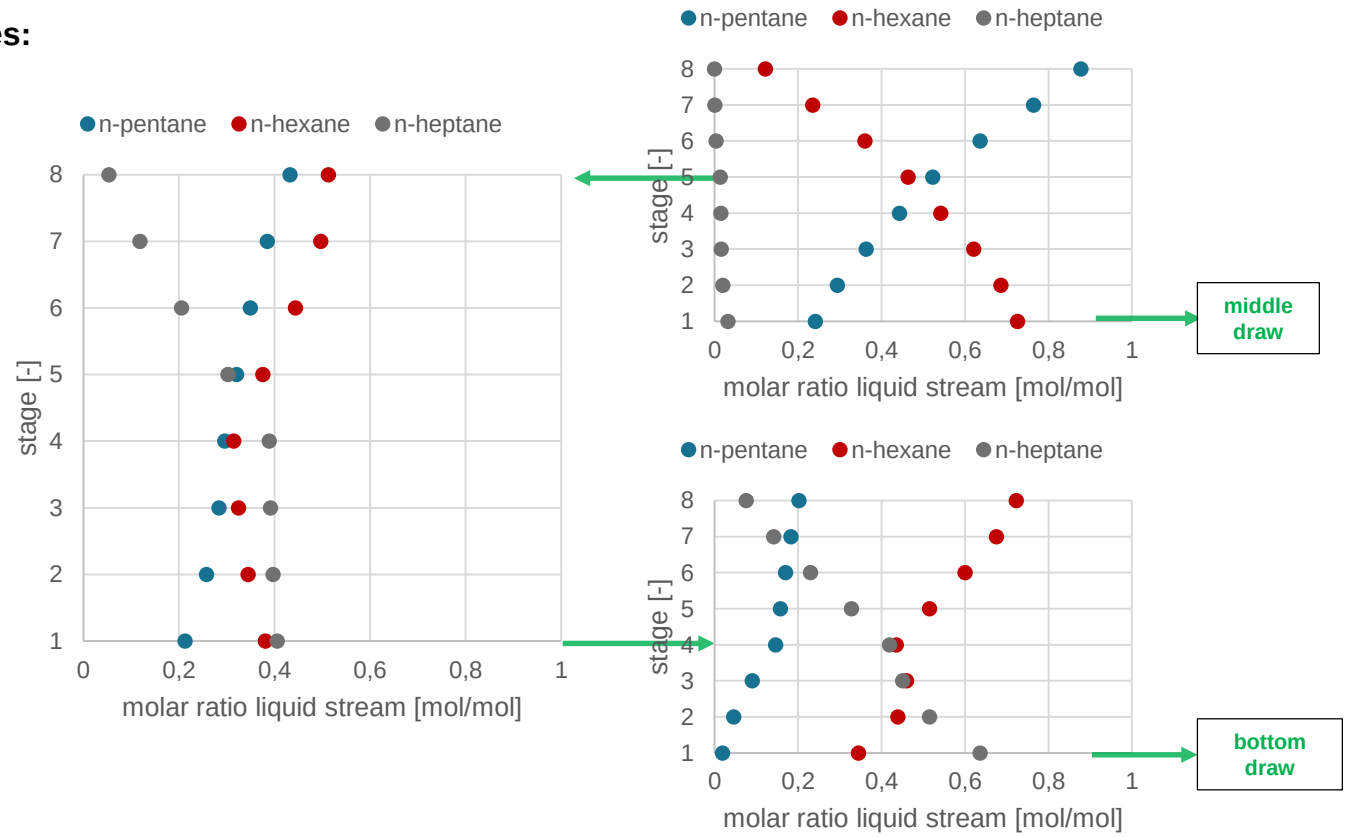
- F = 0,2752 kmol/s
- Composition = {**0.958**, 0.042, 0.000}

Middle draw:

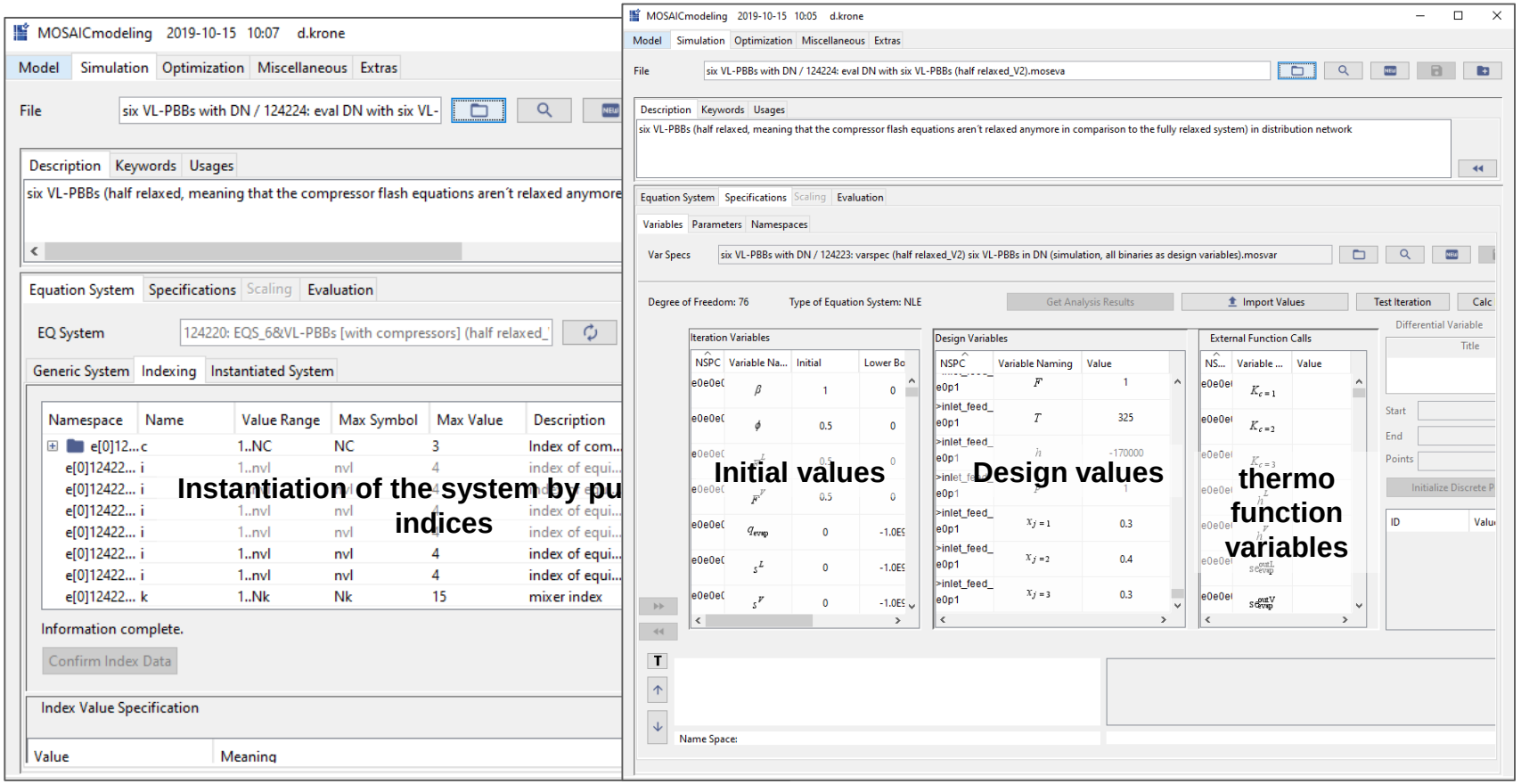
- F = 0,1 kmol/s
- Composition = {0.241, **0.726**, 0.032}

Bottom draw:

- F = 0,6236 kmol/s
- Composition = {0.019, 0.345, **0.636**}



II. Optimization Workflow
1. Model Development – Instantiation and Initialisation



I. Process synthesis approach

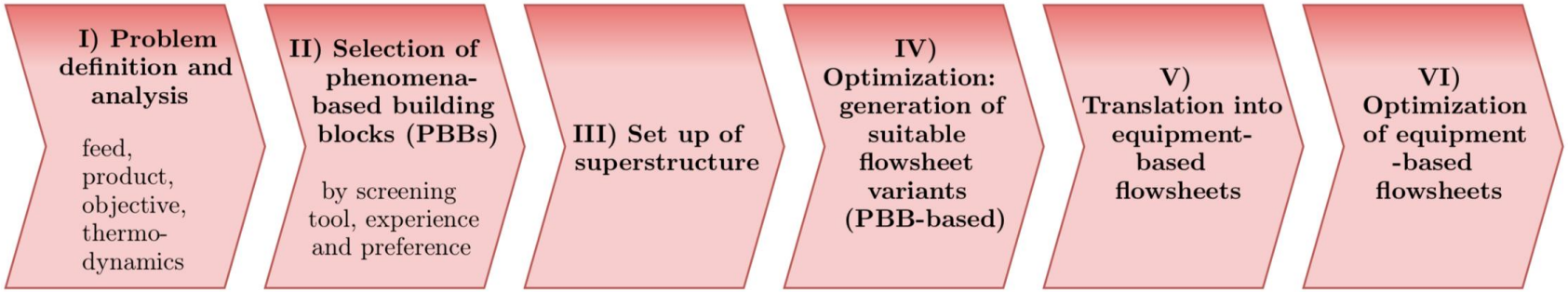


Fig. 1: Process synthesis approach based on the framework by Kuhlmann and Skiborowski[1]

[1] H. Kuhlmann, M. Skiborowski, Ind. Eng. Chem. Res. 2017, 56 (45), 13461 - 13481