Virtual Upscaling

Model Transformation (HSC-Sim -> SULCA)

KASHIF GULZAR(KG), AALTO UNIVERSITY & REINO RUUSU(RR), VTT.

ANNUAL VIRTUAL UPSCALING WEBINAR

20.12.2017

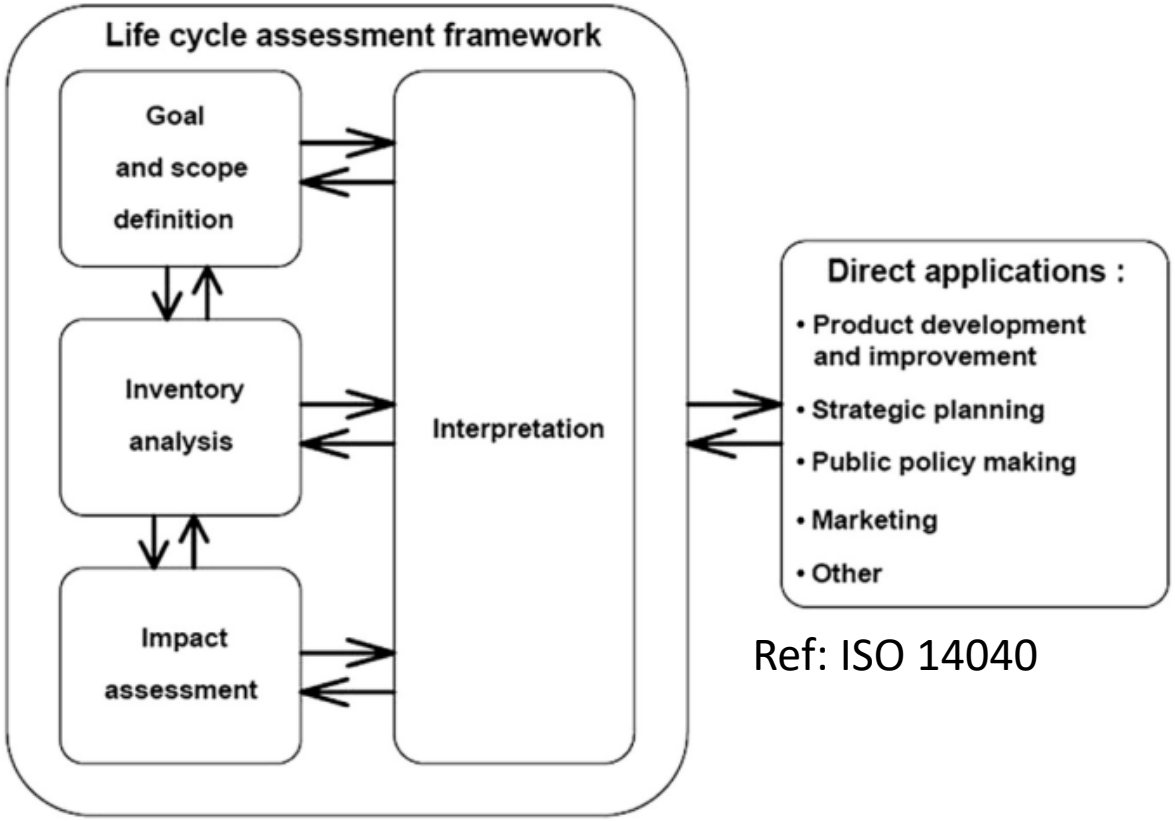
Content

- Introduction
- Model Transformation Architecture
- CHR Rules & SCL Functions
- Transformation Example from HSC-SIM to SULCA (SO_2 Oxidizer, Hydro example 1, Hydro Example2)
- Conclusions
- Questions

LCA Introduction

The concept of LCA is portrayed as the compilation and evaluation of inputs, outputs and potential environmental impacts of a product system (product/service) throughout its life cycle (ISO 14040).

Life cycle stage, including resource extraction, production, transportation, use/consumption, end-of-life activities (collection, sorting, recycling, waste disposal) should to be acknowledged and included in an LCA.



Ref: ISO 14040

Ref: Outotec sim9 help

Why transformation?

- Cradle to Gate (Raw materials extraction and production, exclude transportation to customer)
- Gate to Gate (One process in the production chain)
- Life cycle inventory (LCI)

  This phase is usually the most time-consuming phase, where the input and output data of the system are studied and collected. LCI answers the question: How much of everything flows where?
- It is good to check the data validity during the LCA process.

Model transformation architecture

HSC Sim source Model

XML exports

Process
Variables
Stream connections

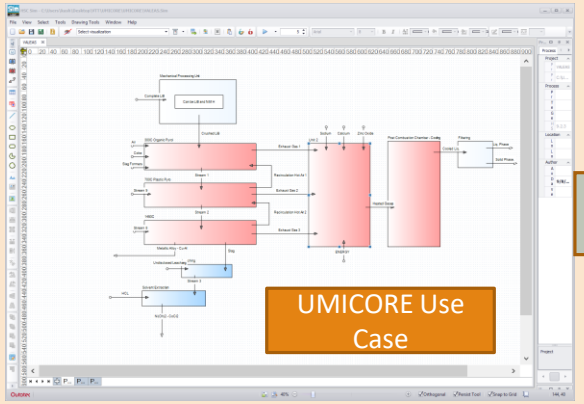
SCL Transformation Mappings

Mass balance calculations from input and output flows

Sulca Model for LCA

LCA Analysis In Sucla

Process created in Sim 9 by Outotec



Process in Sucla perform LCA
Calculates GWP100, Sources of CO_2, and water footprints

CHR Rules

- **Constraint Handling Rules** (**CHR**) a declarative, rule-based language,
- A CHR program, sometimes called a *constraint handler*, is a set of rules that maintain a *constraint store*, a multi-set of logical formulas.
- Execution of rules may add or remove formulas from the store, thus changing the state of the program.

Rule 0	Create multipage flowsheets
Rule 1	Extract Variables
Rule 2	Create SULCA processes blocks
Rule 3	Create Input/output and map associated variables
Rule 4	Create connections
Rule 5	Find loops and make dummy processes for solvable SULCA model

```
// Stream has no source, create dummy process
    when HSCStream ?stream
        not $ existsStatement ?stream H.Stream.HasSource
        ?name = "\(nameOf ?stream)_source"
        ?label = "\(relatedValue ?stream L0.HasLabel :: String)"
        ?process = createProcessInFlowsheet ?name ?label fs
        ?graphics = (?stream # H.Stream.HasGraphics)!0
        ?pageNum = relatedValue ?graphics H.Graphics.HasPage :: Integer
        ?points = relatedValue ?graphics G2D.HasPoint2DArray :: [Double]
        ?x = ?points!0
        ?y = ?points!1
        ?bounds = [?x - dummyWidth / 2, ?y - dummyHeight / 2, dummyWidth,
dummyHeight]
    then print "Input stream: \(?label)"
        Page ?pageNum
        ApplyUnitBounds ?process ?pageNum ?bounds
        StreamSource ?stream ?process
        InputProcess ?process
        UnitBounds ?process ?pageNum ?bounds
        ConfigureProcess ?process ?name ?label
```

SCL Function Call For Model Transformation

The SUCLA model can be generated using the following function in scl.

```
createSulcaModelFromHSC hscModel allowedUnits allowedVariables variableMappings streamVariable
```

Example.

```
createSulcaModelFromHSC (resource "http://HSCSim@A/S2mass") ["t/h", "kW"] [] ([]) "Amount"
```

Parameters and arguments in SCL function call.

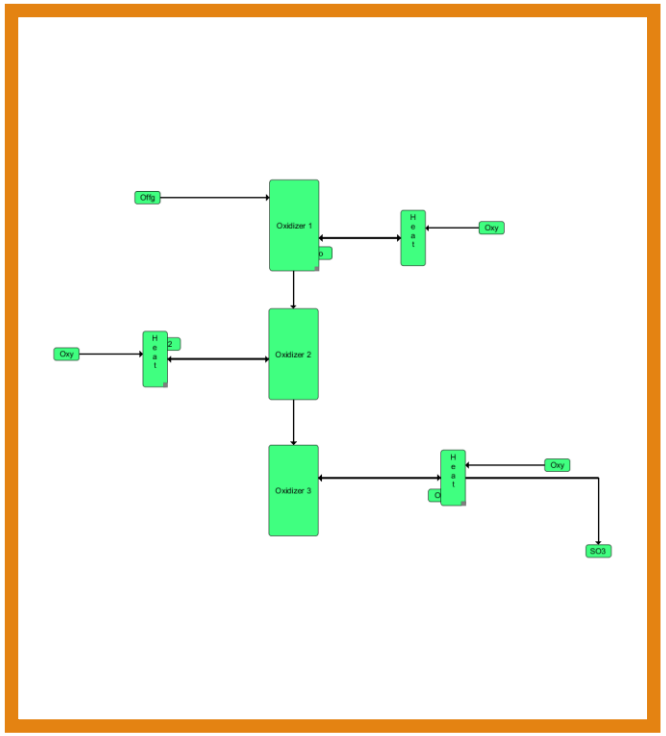
Argument_1-> hscModel : resource in memory for imported xml file

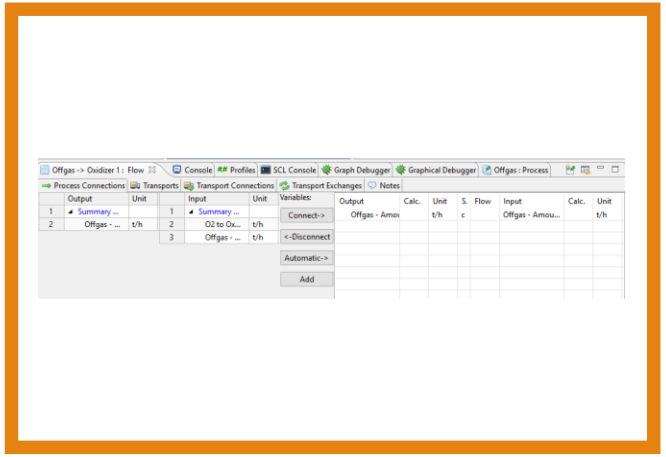
Argument_2-> `allowedUnits` : list of strings of the allowed unit for the transformation

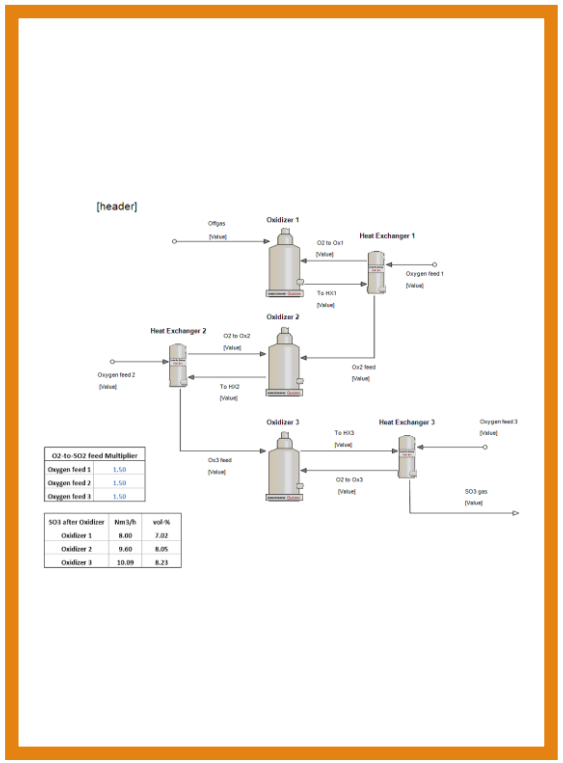
Argument_3-> `allowedVariables`: list of variable names used in the transformation.

Argument_4-> `allowedVariables`: argument allowed for mapping of a list of HSC-Sim variable names to a list of new variable names in SULCA.

Argument_5-> `streamVariable`: stream variables used for making connections between processes.

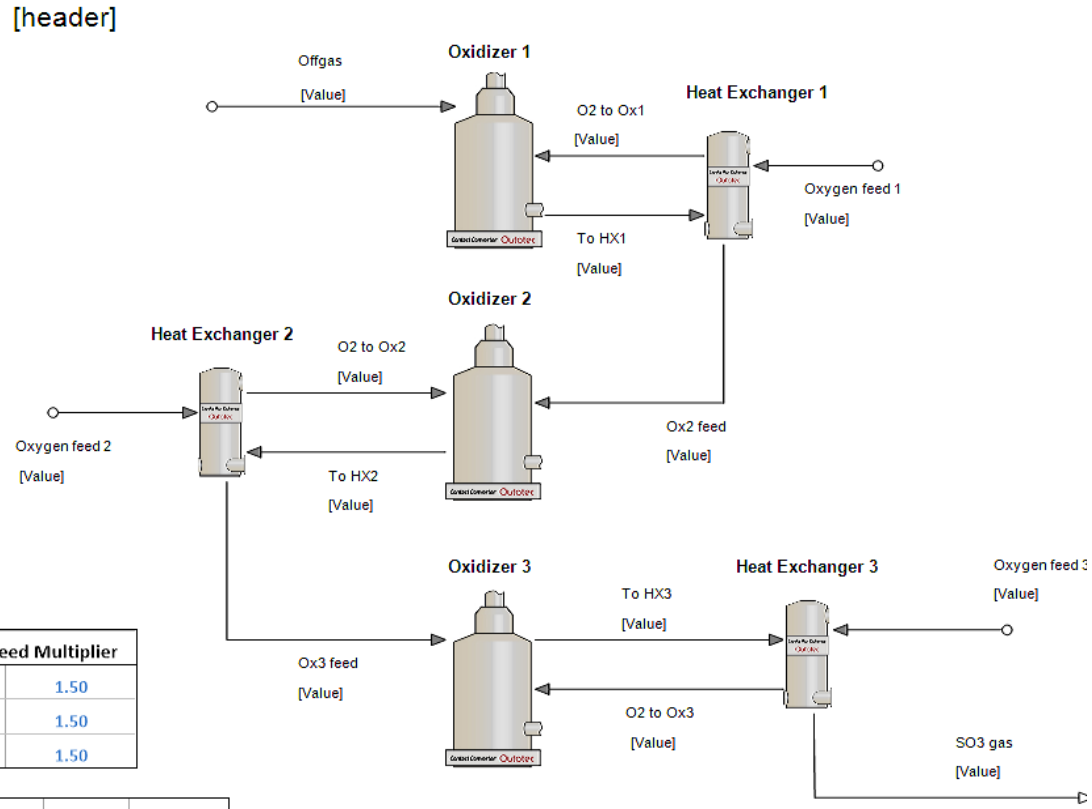






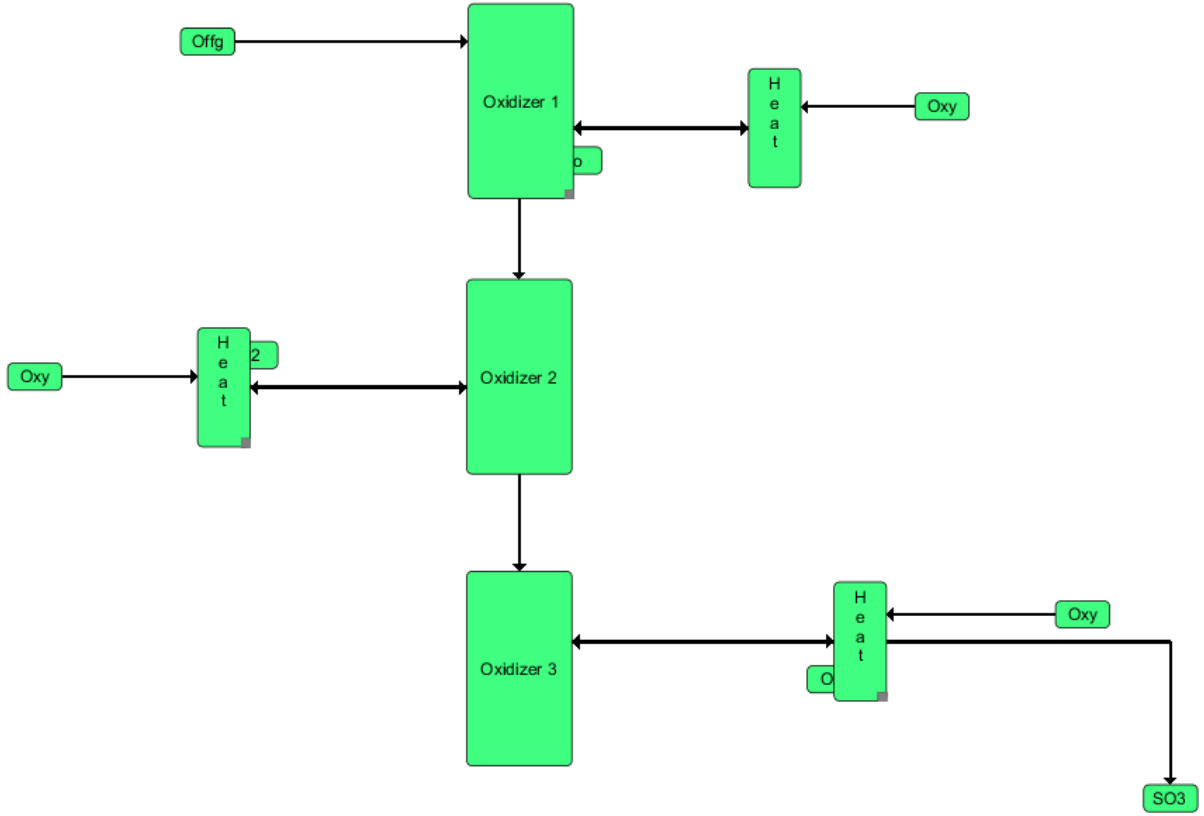
Transformation Examples

SO2 Oxidizer Example



O2-to-SO2 feed Multiplier	
Oxygen feed 1	1.50
Oxygen feed 2	1.50
Oxygen feed 3	1.50

SO3 after Oxidizer	Nm3/h	vol-%
Oxidizer 1	8.00	7.02
Oxidizer 2	9.60	8.05
Oxidizer 3	10.09	8.23



Input Variable mappings and connections

Offgas -> Oxidizer 1 : Flow | Console | Profiles | SCL Console | Graph Debugger | Graphical Debugger | Offgas : Process

Process Connections | Transports | Transport Connections | Transport Exchanges | Notes

	Output	Unit		Input	Unit
1	Summary ...		1	Summary ...	
2	Offgas - ...	t/h	2	O2 to Ox...	t/h
			3	Offgas - ...	t/h

Variables: Connect-> | <-Disconnect

Output	Calc.	Unit	S.	Flow	Input	Calc.	Unit
Offgas - Amou		t/h	c		Offgas - Amou...		t/h

Offgas : Process | Console | Profiles | SCL Console | Graph Debugger | Graphical Debugger | Offgas : Process

Selection from editor 'S2mass'

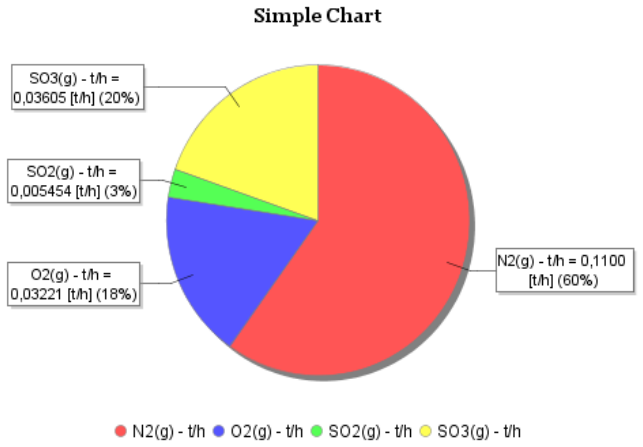
General | Exchanges | DataSheet | Equations | Parameters | Junctions | Impact Assessment

	Input Variable	Amount	Calculated	Unit		Output V...	Amount	Calculated	Unit	Quantity
1	Elements				1	Sum...				
2	N (element) - t/h	0.1100		t/h	2	Off...	0.1443		t/h	t/h
3	O (element) - t/h	0.01713		t/h						
4	S (element) - t/h	0.01717		t/h						
5	Species									
6	N2(g) - t/h	0.1100		t/h						
7	SO2(g) - t/h	0.03430		t/h						
8	Summary Variables									
9	Enthalpy - kW	-28.62		kW						
10	Exergy - kW	53.00		kW						
11	Heat Content - kW	15.52		kW						
12	Offgas - Amoun...	0.1443		t/h						

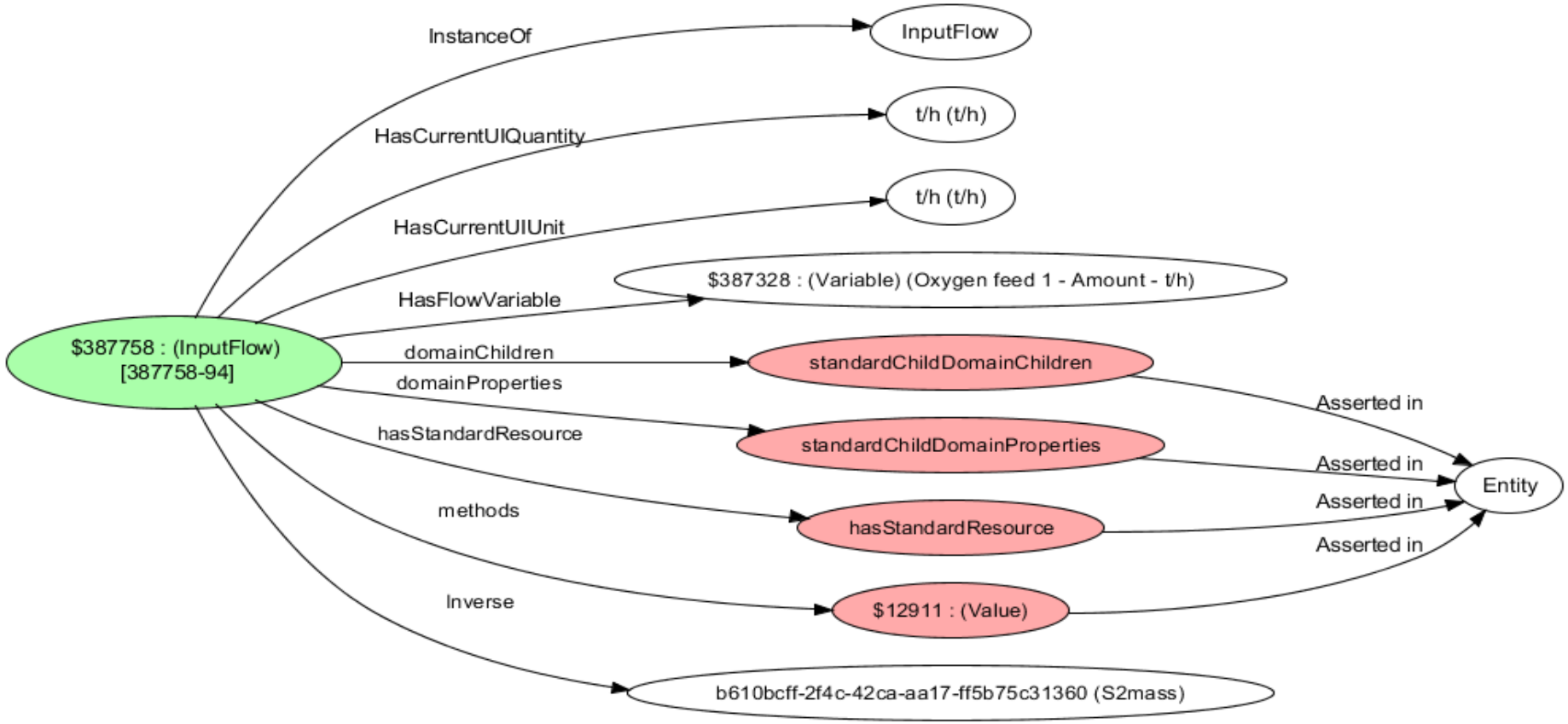
Add | Delete | Replace | Equation | Func. | Ref. | View

Variables

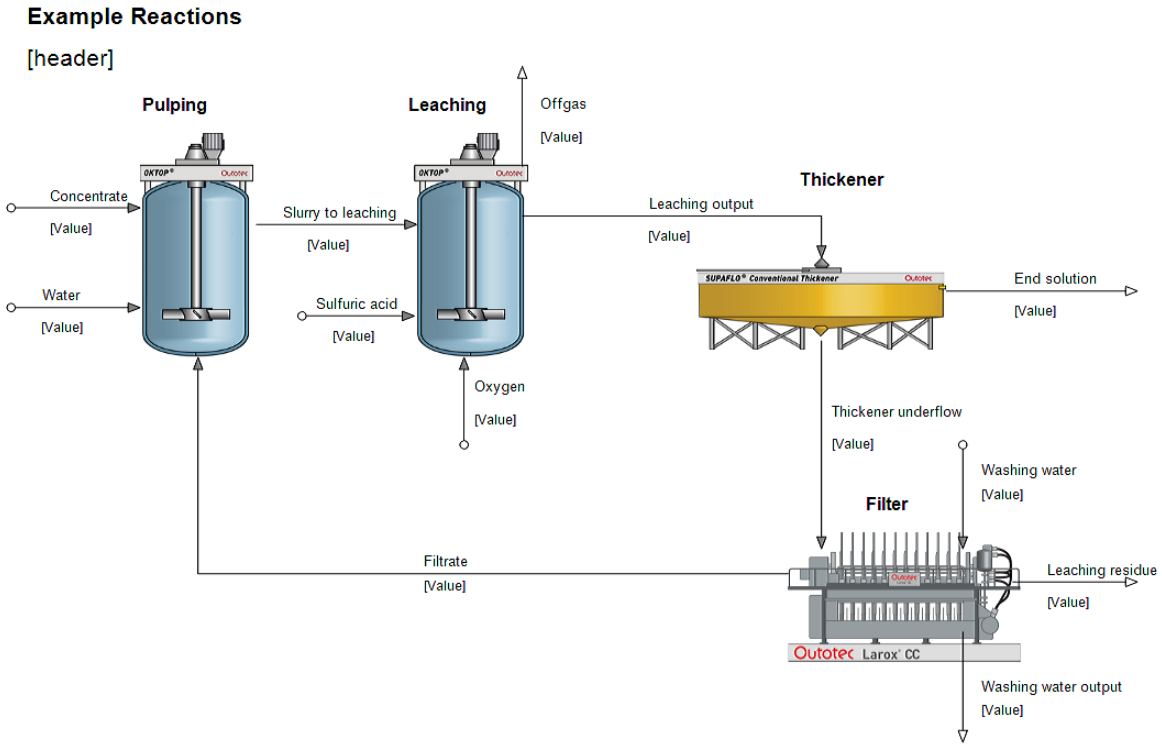
Connection

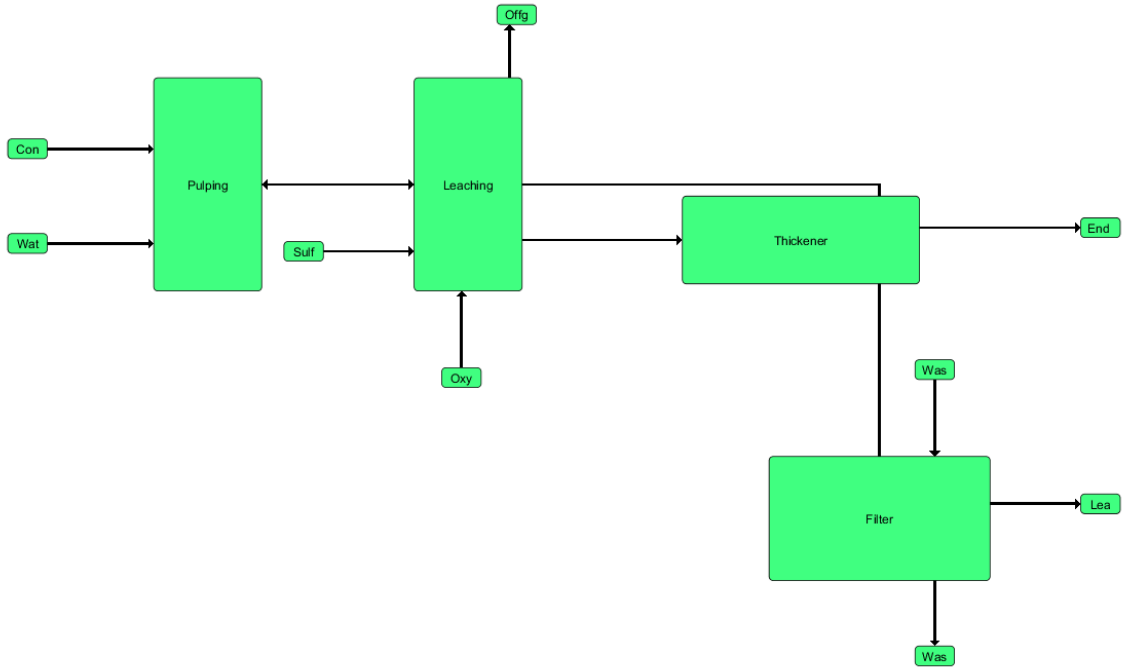


Flow ontology Graph

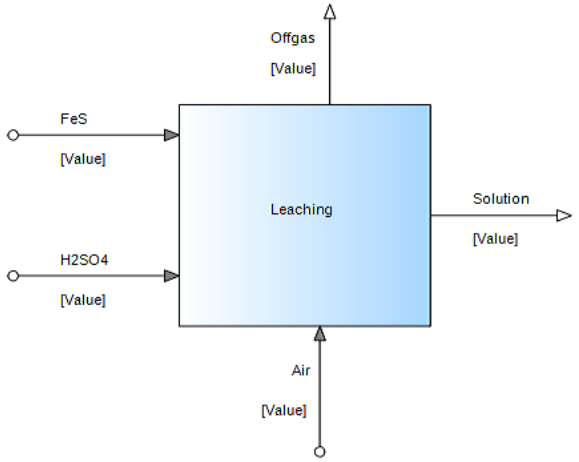


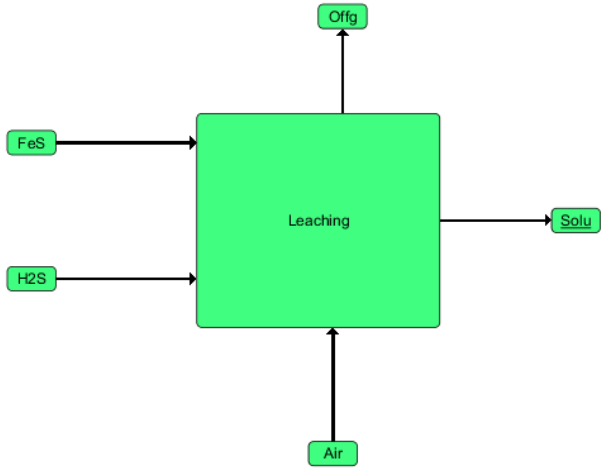
Hydro Example

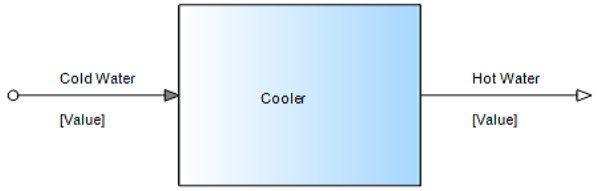


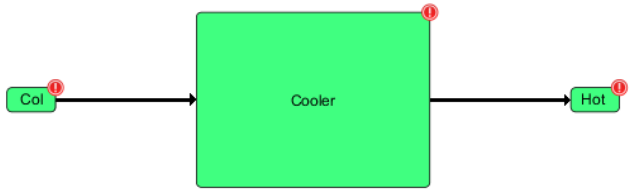


Hydro Example 2 Unconnected blocks Transformed results









Conclusion and Current progress

HSC Sim model Imports library in Sulca (RR).

HSC Process to SULCA Process transformation and IO variable mapping using SCL scripts (RR).

Streams connection and IO mappings and transformations in SCL are completed(RR, KG).

Mass balancing is automatic, a balance system is transformed already available in HSC-Sim which is verified.

Heat and other variable parameters can be selectively made available in the SULCA to perform energy consumptions.

Questions

References

[1] Sim 9: https://www.outotec.com/products/digital-solutions/hsc-chemistry/

[2] SULCA - Sustainability tool for Ecodesign, Footprints & LCA: https://www.simulationstore.com/sulca

[3] Simantics Developer Documentation: http://dev.simantics.org/index.php/Main_Page

[4] CHR Guide: http://www.simantics.org/~niemisto/CHRGuide.html