Transition to CAPE-OPEN COBIA in SINTEF

CO-LaN – CAPE-OPEN 2021 Annual Meeting

Olaf Trygve Berglihn, Senior Research Scientist

Torbjørn Pettersen, Senior Project Manager

Bernd Wittgens, Senior Advisor

2021-10-27

Outline

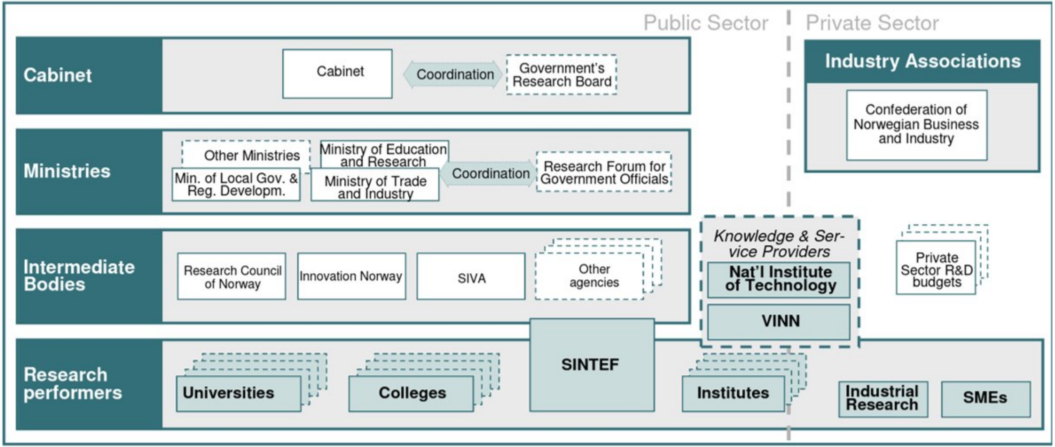
- Bio
- SINTEF: a not-for-profit research foundation
- Our history with CAPE-OPEN in NTNU and SINTEF
- COBIA to the rescue
- Our current work on COBIA adoption
- Some feedback and a wish-list

Bio

- PhD in chemical engineering modelling and thermodynamics
- Senior Research Scientist, with SINTEF since 2008
- 3 years in industry (Linux/IT-company) and medical informatics – long time ago
- 2021-2022 - Visiting research scholar at Renewable and Sustainable Energy Institute, CU Boulder, Colorado, USA

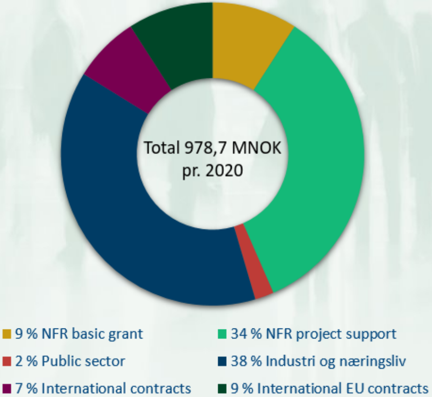
On the 36 bike highway towards Boulder, Colorado

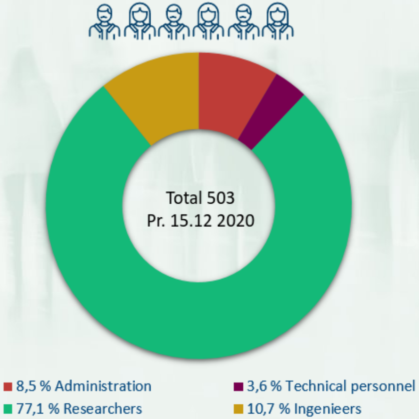
The Norwegian research scene



Public Sector
Private Sector
Cabinet
Cabinet
Coordination
Government's Research Board
Industry Associations
Confederation of Norwegian Business and Industry
Ministries
Other Ministries
Ministry of Education and Research
Min. of Local Gov. & Reg. Developm.
Ministry of Trade and Industry
Coordination
Research Forum for Government Officials
Intermediate Bodies
Research Council of Norway
Innovation Norway
SIVA
Other agencies
Knowledge & Service Providers
Nat'l Institute of Technology
VINN
Private Sector R&D budgets
Research performers
Universities
Colleges
SINTEF
Institutes
Industrial Research
SMEs

SINTEF Industry, not-for-profit RTO - some basic figures

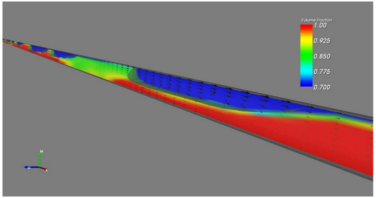


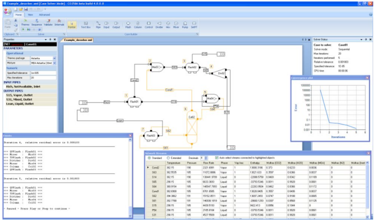


35,4 % of our employees are from abroad,
and from 50 different nations.

Our history with CAPE-OPEN

- Exposure to CAPE-OPEN plans at early phase when at NTNU as MSc-student in late 1999. NTNU contribution to the Global CAPE-OPEN EU-project never materialized (or so I think).
- SINTEF became associate member in 2005 - largely dormant membership
- Leda-project and CO2SIM - CAPE-OPEN was considered but never adopted
- COBIA - now things become interesting. First attempts at implementing with COBIA in real projects started spring 2021





Why did we not adopt CAPE-OPEN before?

- COM and CORBA - complicated, code-intrusive, the Microsoft ecosystem and ATL
- Overly object-oriented focus
- Very much platform dependent

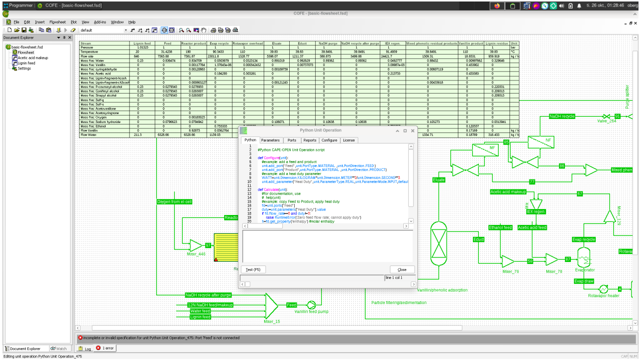
```
/* Opaque pointer */
typedef struct F *F;

/* Functional interface */
void fluid_get_density(F *fluid, double T, double P, double n[], double
void fluid_get_enthalpy(F *fluid, double T, double P, double n[], double
void fluid_get_diffusivity(F *fluid, double T, double P, double n[], dou
void fluid_get_fugacity(F *fluid, double T, double P, double n[], double
/* (...) */
void fluid_free(F *fluid);
```

Simple functional interface in one of our simulators. Binary compatible, C-ABI.

What does COBIA solve for SINTEF

- We work increasingly with Linux - platform independence. Why not also Mac OS?
- COBIA removes need to deal with COM/CORBA
- We’re still compatible with COM and can deliver on the commercial simulator platforms on Windows



Stream	Lignin feed	Feed	Reactor product	Evap recycle	Rotavapor overhead	Eluate	Educt	NaOH purge	NaOH recycle after purge	IEX regen.	Mixed phenolic residual products	Vanillin product	Lignin residue	Unit
Pressure	1.01325	1	1	1	1	1	1	1	1	1	1	1	1	bar
Temperature	20	31.4236	190	90.3433	110	39.85	39.85	39.8491	39.8491	91.4909	39.8491	110	39.85	°C
Flow rate	846	7565.88	7581.87	23011.8	1328.77	5398.07	1211.37	388.875	3499.88	24421.7	1509.31	18.8331	959.919	kg / h
Mass frac Water	0.25	0.836474	0.834709	0.0503679	0.0325134	0.891019	0.992629	0.89362	0.89362	0.0492777	0.88432	0.00997662	0.329646	
Mass frac Vanillin	0	0	0.00117764	1.57643e-06	0.000542432	0	0.00737075	0	0	3.09987e-05	0	0.433902	0	
Mass frac syringaldehyde	0	0	0.00120863	0	0	0.00169759	0	0	0	0	0.00607119	0	0	
Mass frac Acetic acid	0	0	0	0.194295	0.503261	0	0	0	0	0.213735	0	0.435385	0	
Mass frac Lignin-fragment-hcooh	0	0	0	0	0	0	0	0	0	0	0	0	0	
Mass frac Lignin-fragment-ch3cooh	0	0	0.000863227	0	0	0.00121245	0	0	0	0	0.00433618	0	0	
Mass frac P-coumaryl-alcohol	0.25	0.0279545	0.0278955	0	0	0	0	0	0	0	0	0	0.220331	
Mass frac Coniferyl alcohol	0.25	0.0279545	0.0265007	0	0	0	0	0	0	0	0	0	0.209315	
Mass frac Sinapyl alcohol	0.25	0.0279545	0.0265007	0	0	0	0	0	0	0	0	0	0.209315	
Mass frac Sulf-g	0	0	0	0	0	0	0	0	0	0	0	0	0	
Mass frac Sulf-s	0	0	0	0	0	0	0	0	0	0	0	0	0	
Mass frac Acetovanillone	0	0	0	0	0	0	0	0	0	0	0	0	0	
Mass frac Acetosyringnone	0	0	0	0	0	0	0	0	0	0	0	0	0	
Mass frac Oxygen	0	0	0.00165025	0	0	0	0	0	0	0	0	0	0	
Mass frac Sodium hydroxide	0	0.0796623	0.0794942	0	0	0.106071	0	0.10638	0.10638	0	0.105273	0	0.0313841	
Mass frac Ethanol	0	0	0	0.755335	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0.120537	0	
Flow Vanillin	0	0	8.92873	0.0362764	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	8.17169	0	kg / h
Flow Water	211.5	6328.66	6328.66	1159.05	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1334.71	0.18789	316.433	kg / h

Python Unit Operation

Python | Parameters | Ports | Reports | Configure | License

```
#Python CAPE-OPEN Unit Operation script

def Configure(unit):
    #example: add a feed and product
    unit.add_port("Feed",unit.PortType.MATERIAL ,unit.PortDirection.FEED)
    unit.add_port("Product",unit.PortType.MATERIAL ,unit.PortDirection.PRODUCT)
    #example: add a heat duty parameter
    WATT=unit.Dimension.KILOGRAM*unit.Dimension.METER**2/unit.Dimension.SECOND**3
    unit.add_parameter("Heat Duty",unit.ParameterType.REAL,unit.ParameterMode.INPUT,defaul

def Calculate(unit):
    #for documentation, use
    #  help(unit)
    #example: copy Feed to Product, apply heat duty
    fd=unit.ports["Feed"]
    duty=unit.parameters["Heat Duty"].value
    if fd.flow_rate==0 and duty!=0:
        raise RuntimeError('Zero feed flow rate; cannot apply duty')
    h=fd.get_property('enthalpy') #molar enthalpy
```

Test (F5) | Close | line 1 col 1

```
Incomplete or invalid specification for unit Python Unit Operation_475: Port 'Feed' is not connected
```

Editing unit operation Python Unit Operation_475

COFE and Amsterchem Python module running under Linux (with Wine)

Technology for a better society

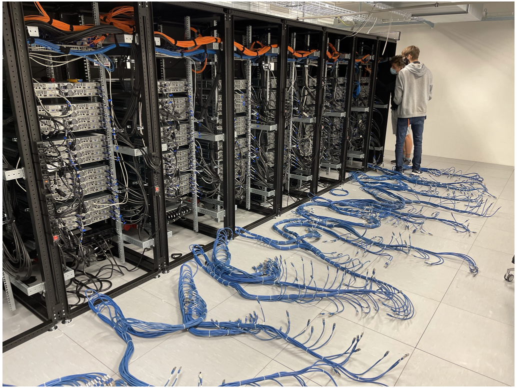
Current activities at SINTEF with COBIA

- Porting novel gas-liquid contactor to COBIA PMC (planning)
- Porting membrane models to COBIA PMC (prototype)
- Porting tailored thermodynamic models for gas absorption (planning)
- Building thermo-models and PMC’s for complex biorefinery processes, lignin depolymerization, etc. (started)

The future for COBIA at SINTEF

We expect increasing use of

- server applications, not desktop
- Linux, not Windows – ideally platform independent code
- binary compatible units and stable interfaces



A section of the SINTEF OpenStack cluster

… and a wish-list

For this we'd like

- A C-ABI, binary compatible, easy to use runtime library – no need to implement all kinds of collections, it's standardized with runtime utility functions
- Language wrappers that all use the same C-ABI – avoid the heavy `C++` dependence and recompile issues
- Prefer runtime functional interfaces over `C++` code generation
- Reduce complexity, keep it simple and stable
- Simulator environments that become platform independent
- Focus on in-proc, leave the out-of-proc to the implementer in an in-proc unit

Acknowledgements

The SINTEF COBIA activity has received funding from the European Union’s Horizon 2020 Research and Innovation programme under the following grant agreements:

- BRISK2: EU H2020 grant agreement 731101
- LIBERATE: EU H2020 SPIRE grant agreement 820735
- SElectiveLi: EU H2020 BBI grant agreement 837276i
- EBIO: EU H2020 grant agreement 101006612