stream

Modelica 3.2

stream prefix

- **Only in a connector declaration**
- **Exactly one “flow” variable**

- **Equations are generated when using the inStream() or actualStream()**

Stream Connector

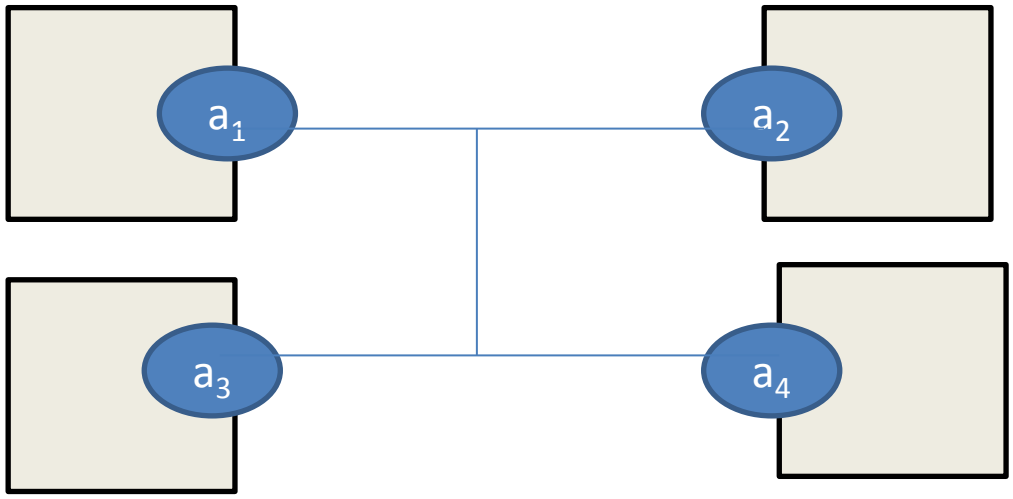
```
connector FluidPort
    Real p(unit="Pa")  "Pressure";
    flow Real q(unit="m3/s")  "Flow into";
    stream Real c(unit="mol/m3")
    "Outflow concentration close to port";
end FluidPort;
```

inStream(a.c)

= mixed concentration in stream

- numerically reliable
- bi-directional transports
- vectorizable
- independent on **a.c** !!!

inStream(a_i.c)



$$0 = \sum_{j=0}^{N} a_j.q \times \begin{cases} c_{mix} & a_j.q \geq 0 \\ a_j.c & otherwise \end{cases}$$

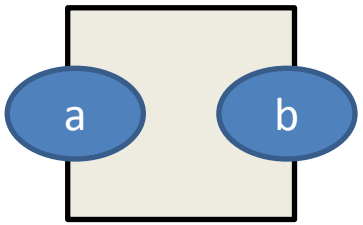
$$inStream(a_i.c) = \begin{cases} c_{mix} & a_j.q \geq 0 \\ ? & otherwise \end{cases}$$

Simple 2-ports component

0 = a.q + b.q;

a.c = inStream(b.c)

b.c = inStream(a.c)



(Resistor, Diode, Inertia, Meassurements,..)

Resistor

0 = a.q + b.q;

a.c = inStream(b.c)

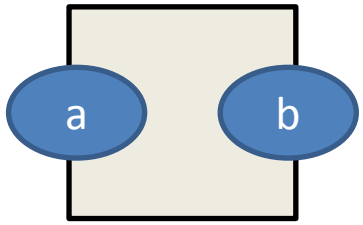
b.c = inStream(a.c)

a.q = (a.p-b.p) / Resistance;

actualStream(a.c)

$$actualStream(a.c) = \begin{cases} inStream(a.c) & a.q > 0 \\ a.c & otherwise \end{cases}$$

ElasticCompartment




```
a.p=(volume-V0)/Compliance +ExternalPressure;
a.p = b.p;
der(volume) = a.q + b.q;
der(soluteMass) = a.q*actualStream(a.c) +
                                  b.q*actualStream(b.c);
a.c = soluteMass / volume;
a.c = b.c;
```