gestion énergétique et modèles réactifs d'occupants dans le bâtiment

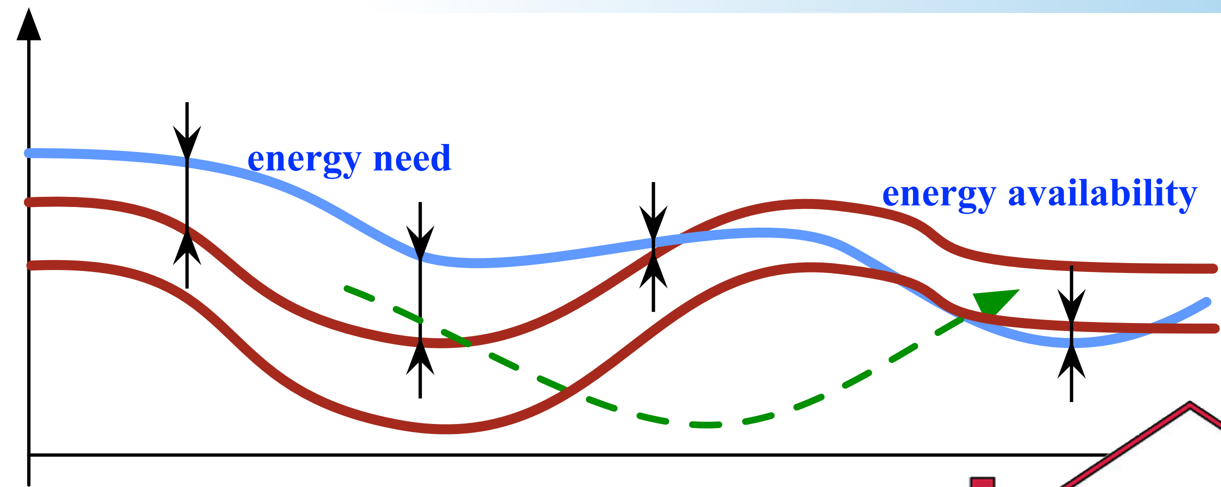
towards a multi-application modeling language for GMBA-BEMS

stephane.ploix@grenoble-inp.fr

Laboratoire G-SCOP
46, av Félix Viallet
38031 Grenoble Cedex
www.g-scop.inpg.fr

journées thématiques IBPSA, Grenoble, jeudi 16 mai 2013 :
le bâtiment vu comme un noeud énergétique à l'interface des usagers et des réseaux

context



intermittence

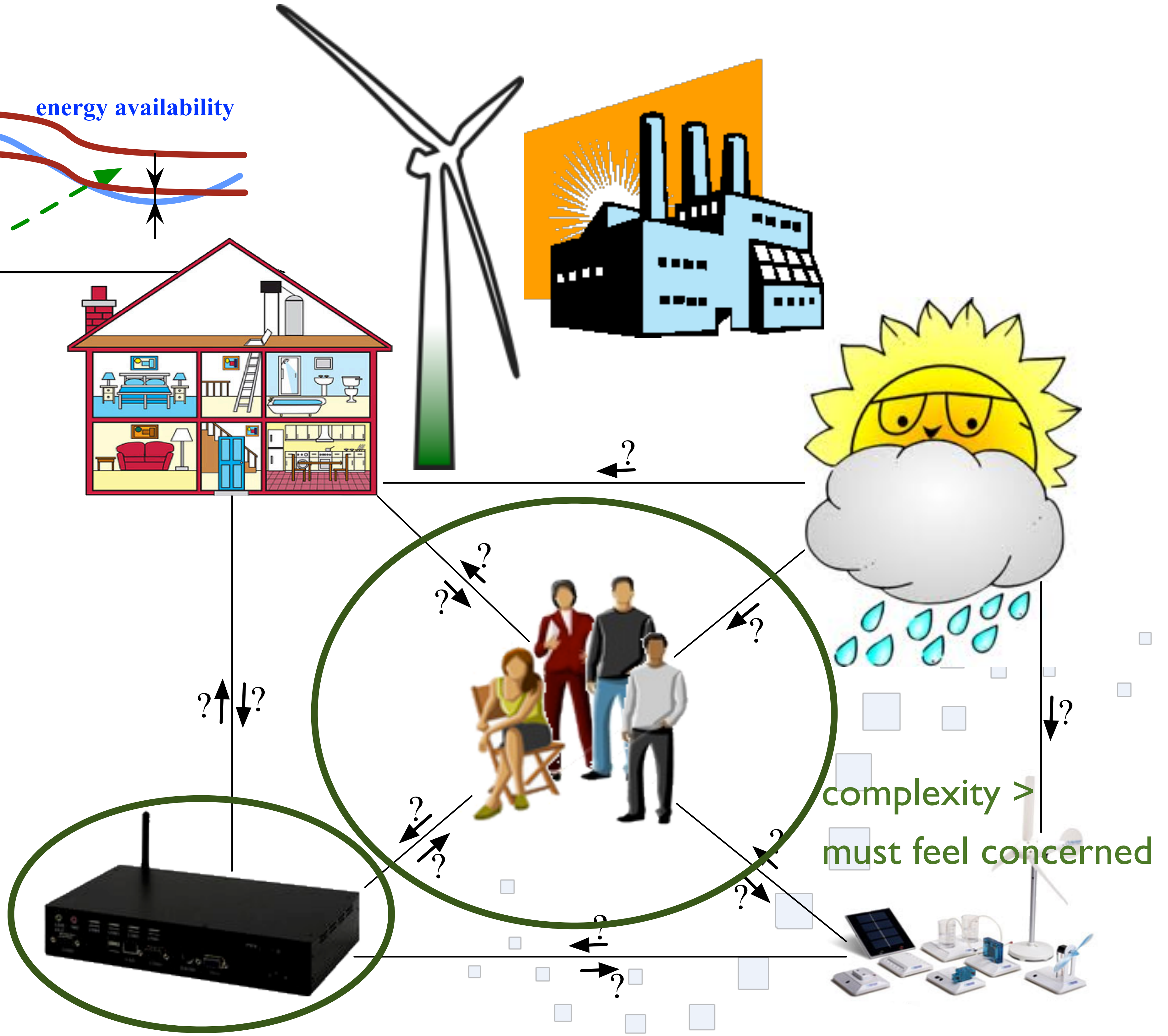
sobriety

diversification

communities

heating network

smart-grid

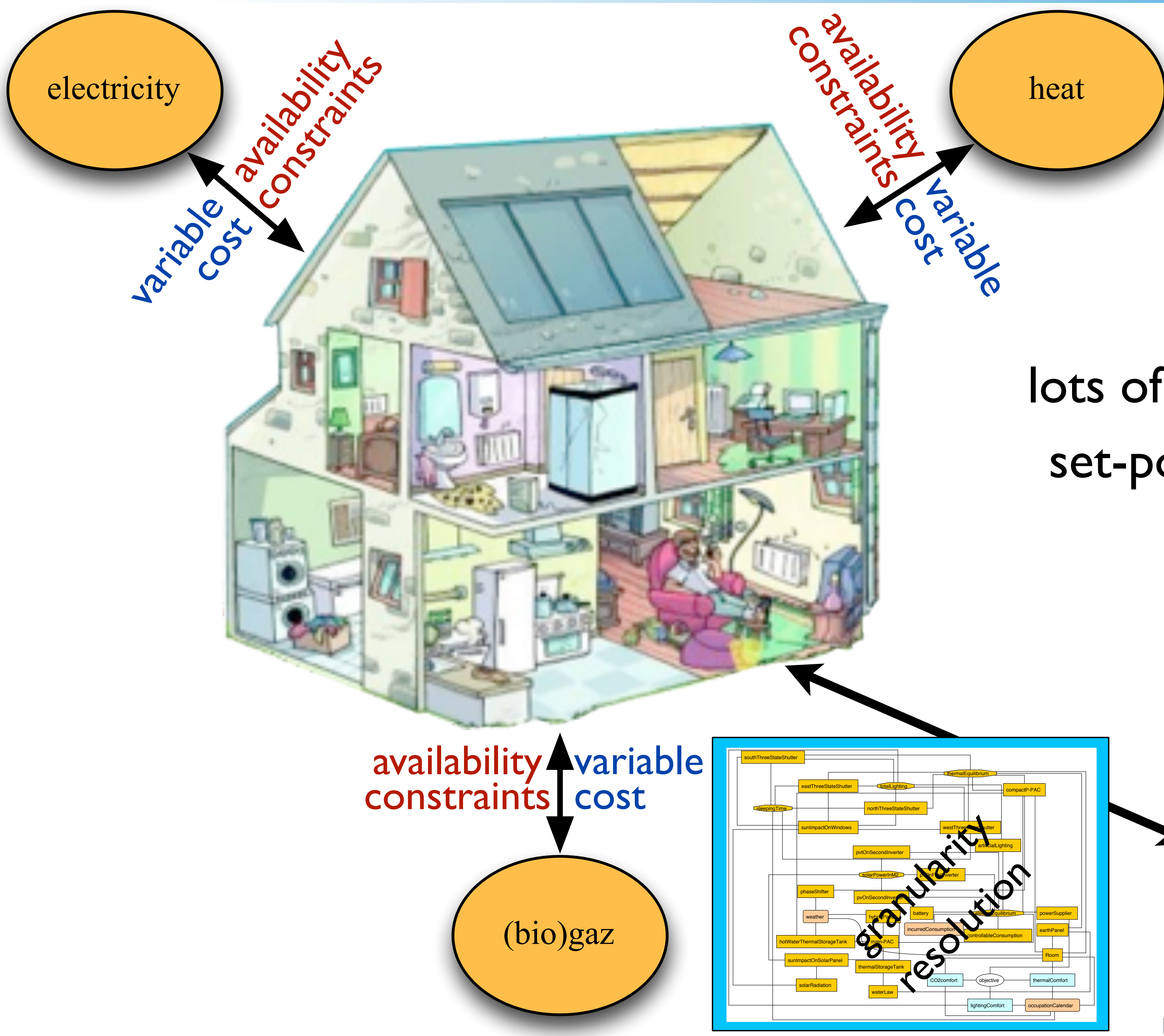


outlines

- why GMBA-BEMS?
- neutral modeling language requirements
- multi-application modeling process
- application example
- modeling occupants

why GMBA-BEMS?

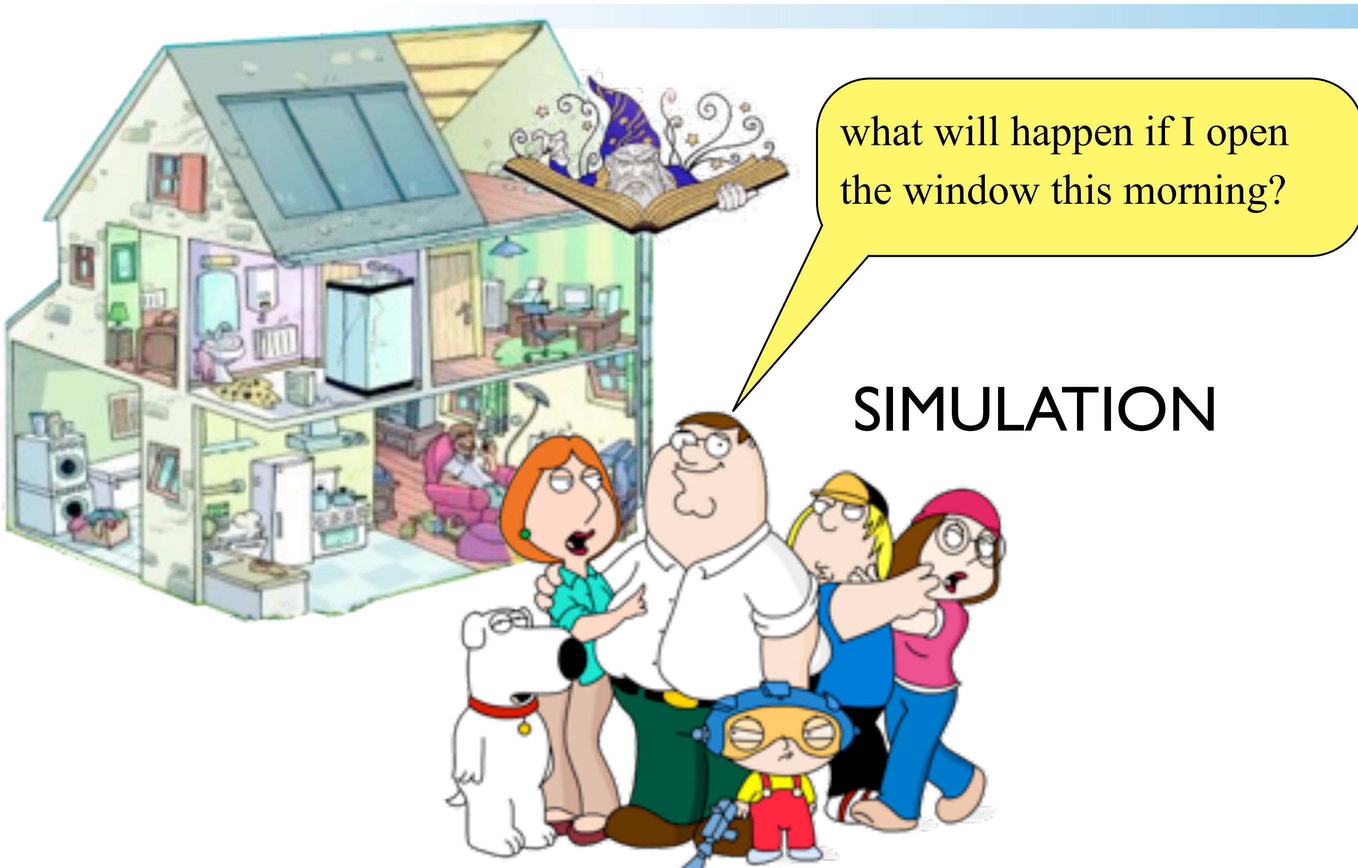
global model based anticipative energy management and models



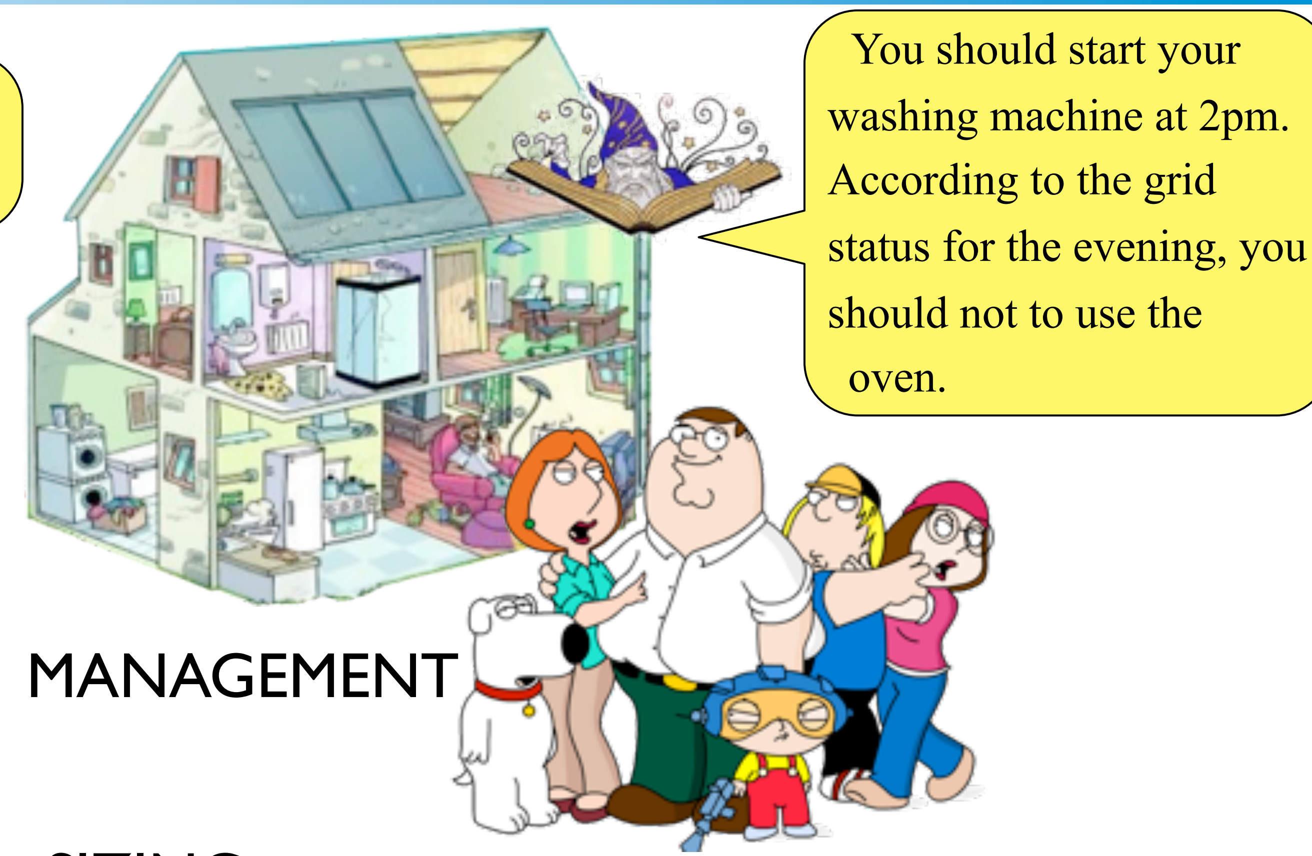
model of the building system

lots of possible configurations, set-points and starting times

models have to be used by different applications

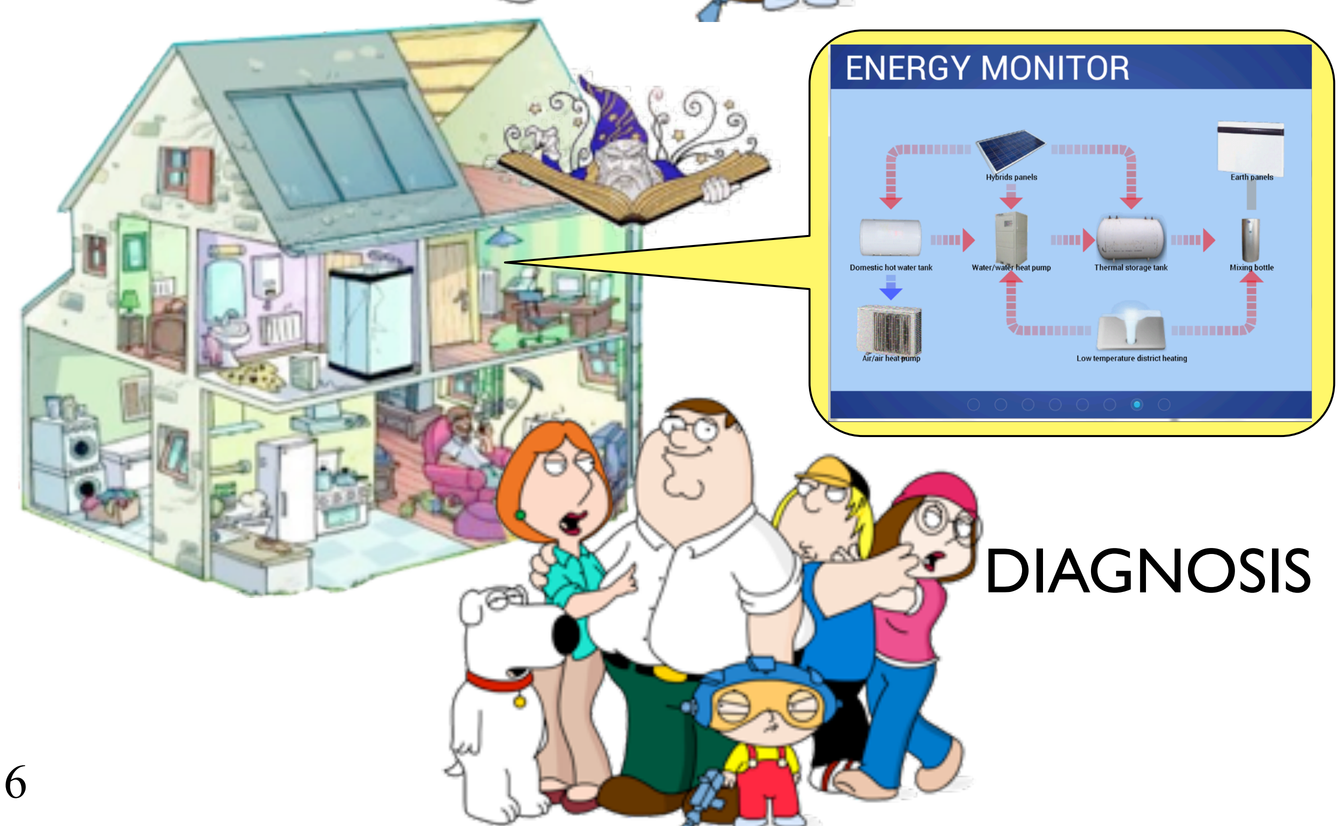


SIMULATION

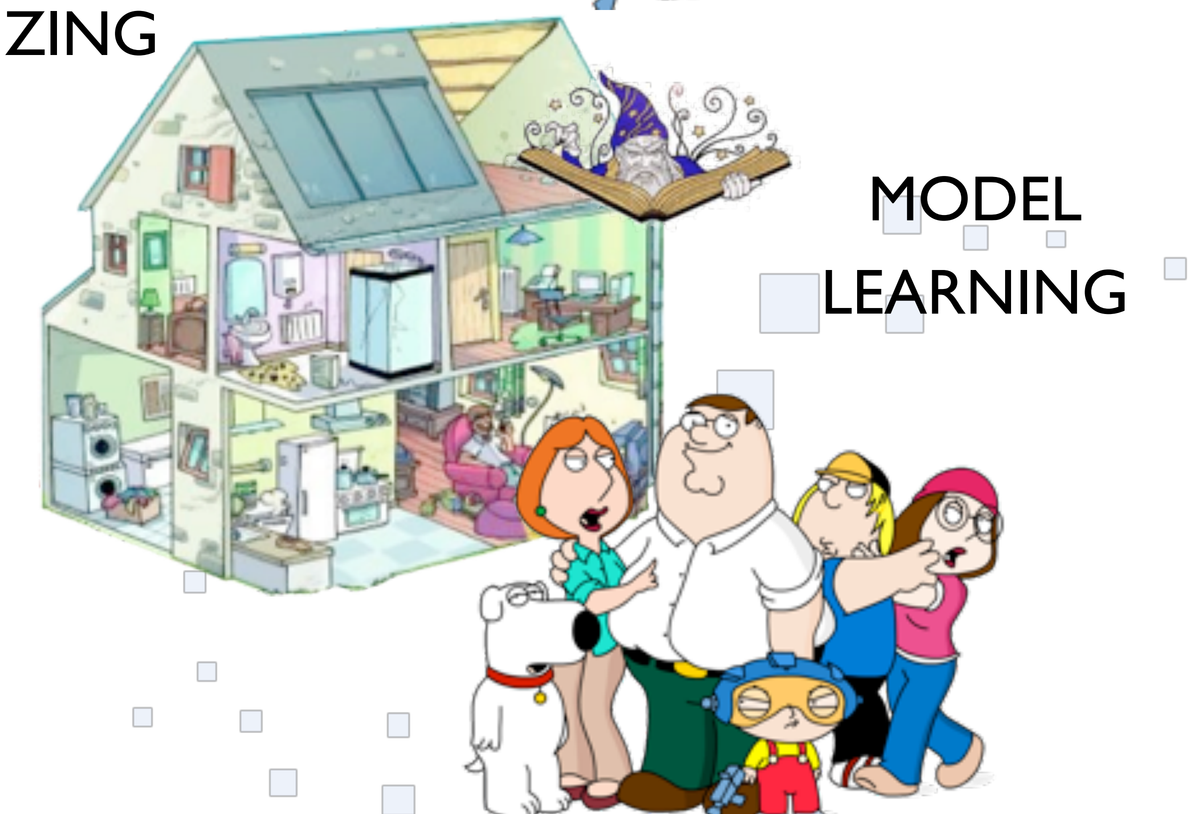


MANAGEMENT

SIZING

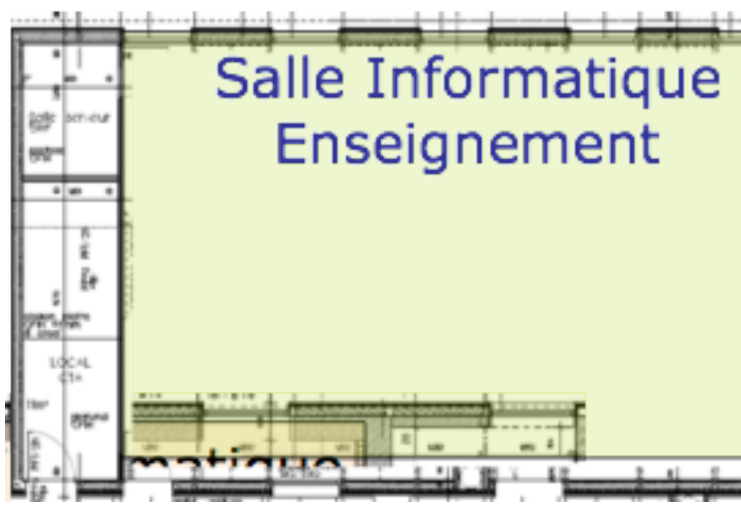


DIAGNOSIS



MODEL LEARNING

models for energy management are difficult to handle



PREDIS/MHI:

1151 continuous variables

144 binary variables

3443 constraints

Armadillo:

700 continuous variables

155 binary variables

2355 constraints

Canopea:

2249 continuous variables

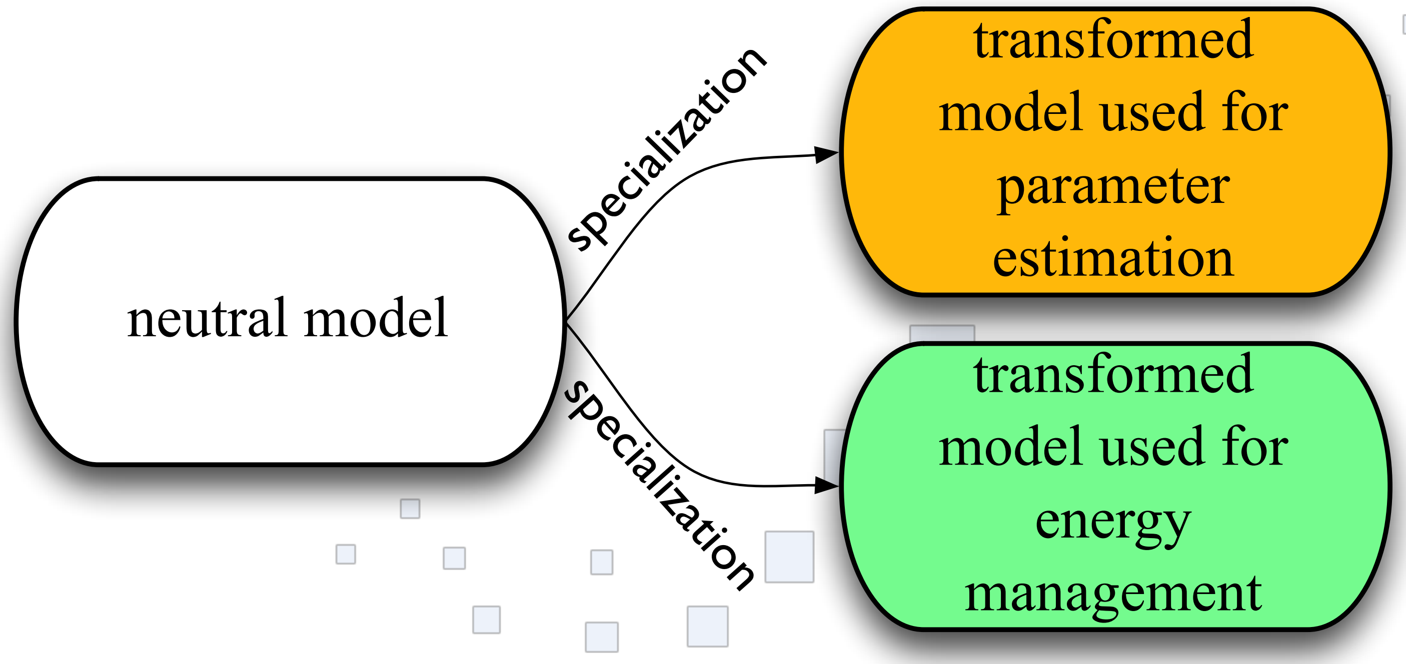
652 binary variables

10168 constraints

neutral modeling language requirements

how to make modeling more efficient?

- re-use model elements for different applications
  - sizing
  - simulation
  - parameter estimation
  - energy management
  - diagnostic analysis
- develop a neutral modeling language and "automatically" transform for applications
  - intantiate and compose
  - demultiply (time,...)
  - linearize (if necessary)



what a neutral model is made of?

- a neutral model is a constrained space
  - variables: symbols with general value domains (usually $\mathbb{R}, \mathbb{R}^+$)
    - isolated or indexed time-invariant variables: $V(i,j) \in dom(V(i,j))$
    - isolated or indexed time-varying variables: $V(i,j,t) \in dom(V(i,j,t))$
  - constants: symbols with single value domain
    - isolated or indexed time-invariant constants: $V(i,j) = \varsigma_{i,j}$
    - isolated or indexed time-varying variables: $V(i,j,t) = \varsigma_{i,j,t}$
  - constraints: physics and occupants' requirements
    - acausal: equality or inequality with acausal logical operator
    - with/without time-varying variables: ODE,..

RoomCalendar

```
tvar occupancy in [0,inf]
tvar presence in {0,1}
presence = 1 equiv occupancy >= 0.1
```

variables may be transformed to constants (i.e. parameters)

are neutral modeling languages existing?

- causal languages cannot be candidate > acausal
  - Matlab/Simulink, TRNSYS,...
- non composable languages cannot be candidate > composable
  - OPL, GMPL, AMPL...
- simulation oriented languages cannot be candidate > constraint satisfaction problem compatible (value domains, inequalities, under/just/over-determined problems)
  - Modelica
- language must be based on existing languages

example of neutral models

ThermalZone

```
var nNeighbourhoods in {0,inf}
tvar Tneighbour[nNeighbourhoods] in [0,inf]
tvar Tout in [0,inf]
tvar Twall in [0,inf]
tvar Tin in [0,inf]
tvar ventilationAirFlow in [0,inf]
tvar inZoneHeat in [-inf,inf]
var Rw in [0,inf]
var Rneighbour[nNeighbourhoods] in [0,inf]
var Cwall in [0,inf]
var efficiency in [0,1]
rho_air=1.184
Cp_air=1006
Rventilation = 1/((1-efficiency)*rho_air*Cp_air*0.61*ventilationAirFlow) # tvar
Ac = -(sum(1/Rneighbour)+1/(Rventilation+Rw))/Cw # tvar, sum(...,1)
Bout = 1/(Rventilation+Rw)/Cw # tvar
Bheat = Rventilation/(Rventilation+Rw)/Cw # tvar
Bneighbour = 1/Rneighbour/Cw # var
C = Rventilation/(Rventilation+Rw) # tvar
Dout = Rw/(Rventilation+Rw) # tvar
Dheat = (Rw*Rventilation)/(Rventilation+Rw) # tvar
var TwallInit
Twall{0}=TwallInit
der(Twall)=Ac*Twall+Bout*Tout+sum(Bneighbour*Tneighbour)+Bheat*inZoneHeat
Tin=C*Twall+Dout*Tout+Dheat*inZoneHeat
```

indexed variable

implicit

potentially nonlinear

operator for indexed variables

multi-application modeling process

how to transform neutral model to application model?

- variables are consecutively specialized
  - symbols (composition):
    neutralSymbol > specN.specN-1...spec1.neutralSymbol
  - value domains:
    dom(neutralSymbol) $\supset$ dom(specN.specN-1...spec1.neutralSymbol)
- constraints are transformed according to application
- new variables and constraints may be introduced because of transformation patterns
- application specific elements are added (objective to minimize,...)

transformation of a neutral model into an energy management model

- composition

Two thermal zones

```
import ThermalZone as LivingRoom
import ThermalZone as BedRoom
LivingRoom.nNeighbourhoods=1
BedRoom.nNeighbourhoods=1
LivingRoom.Tneighbour[0]=BedRoom.Tin
BedRoom.Tneighbour[0]=LivingRoom.Tin
BedRoom.Tout=LivingRoom.Tout
```

- specialization

Living room

```
LivingRoom.Rw=1.2e-3
LivingRoom.Rneighbour[0]=4.4e-3
LivingRoom.Cw=1.53e+7
LivingRoom.inZoneHeat in [0,10000]
```

transformation of a neutral model into an energy management model

- transformation: time discretization (Te=3600s, H=24h)

Time discretization (Te=3600,H=24*3600)

```
Tout > Tout{0},Tout{1},...,Tout{22},Tout{23}
der(Twall)=Ac*Twall+Bout*Tout+sum(Bneighbour*Tneighbour)+Bheat*inZoneHeat >
(Twall{1}-Twall{0})/Te=Ac{0}*Twall{0}+Bout{0}*Tout{0}+sum(Bneighbour*Tneighbour{0})+Bheat{0}*inZoneHeat{0}
(Twall{2}-Twall{1})/Te=Ac{1}*Twall{1}+Bout{1}*Tout{1}+sum(Bneighbour*Tneighbour{1})+Bheat{1}*inZoneHeat{1}
...
(Twall{23}-Twall{22})/Te=Ac{22}*Twall{22}+Bout{22}*Tout{22}+sum(Bneighbour*Tneighbour{22})+Bheat{22}*inZoneHeat{22}
```

Linearization

```
ventilationAirFlow in {2/3600,400/3600,800/3
```

Formulation

$$\pi = x \times y; x = [\check{x}, \hat{x}], y = [\check{y}, \hat{y}]$$

transformation

$$y \to y \in \{y_0, \ldots, y_n\}; y_i = \check{y} + i\frac{\hat{y} - \check{y}}{n}$$

It can be modeled by:

$$\begin{aligned} y &= \delta_0 y_0 + \delta_1 y_1 + \cdots + \delta_n y_n \\ \text{with} & \sum_i \delta_i = 1; \delta_i \in \{0, 1\} \end{aligned}$$

It yields:

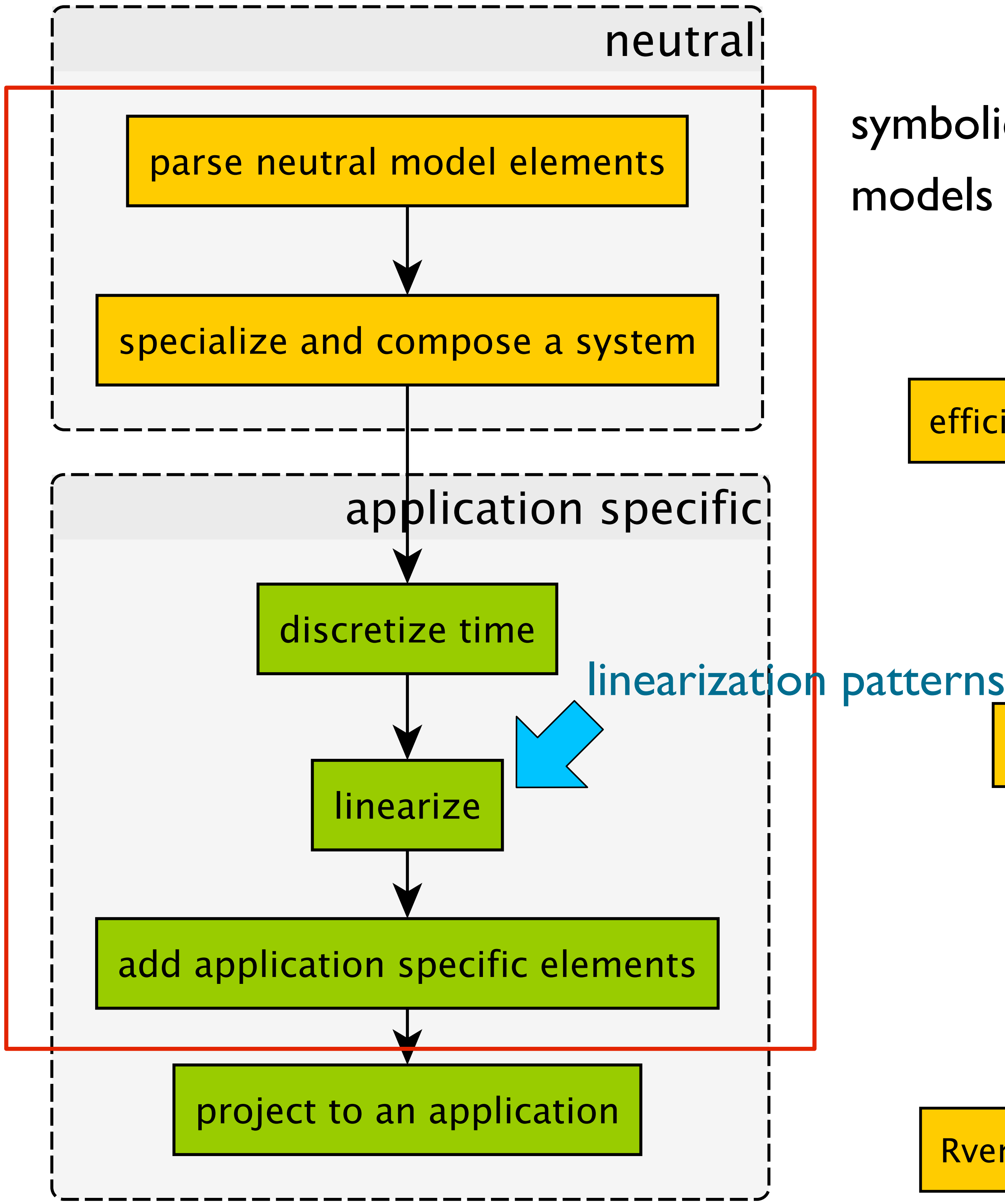
$$\pi' = y_0(\delta_0 x) + y_1(\delta_1 x) + \cdots + y_n(\delta_n x)$$

Let $z_i = \delta_i x$. using the linearization of the product of a binary variable with a continuous one, it yields:

$$\pi' = \sum_{i \in \{0,\ldots,n\}} y_i z_i \; ; \; x = [\check{x}, \hat{x}], y_i = \check{y} + i\frac{\hat{y} - \check{y}}{n}$$

$$\forall i, \begin{cases} z_i \leq & \hat{x}\delta_i \\ z_i \geq & \check{x}\delta_i \\ z_i \leq & x - \check{x}(1 - \delta_i) \\ z_i \geq & x - \hat{x}(1 - \delta_i) \end{cases}$$

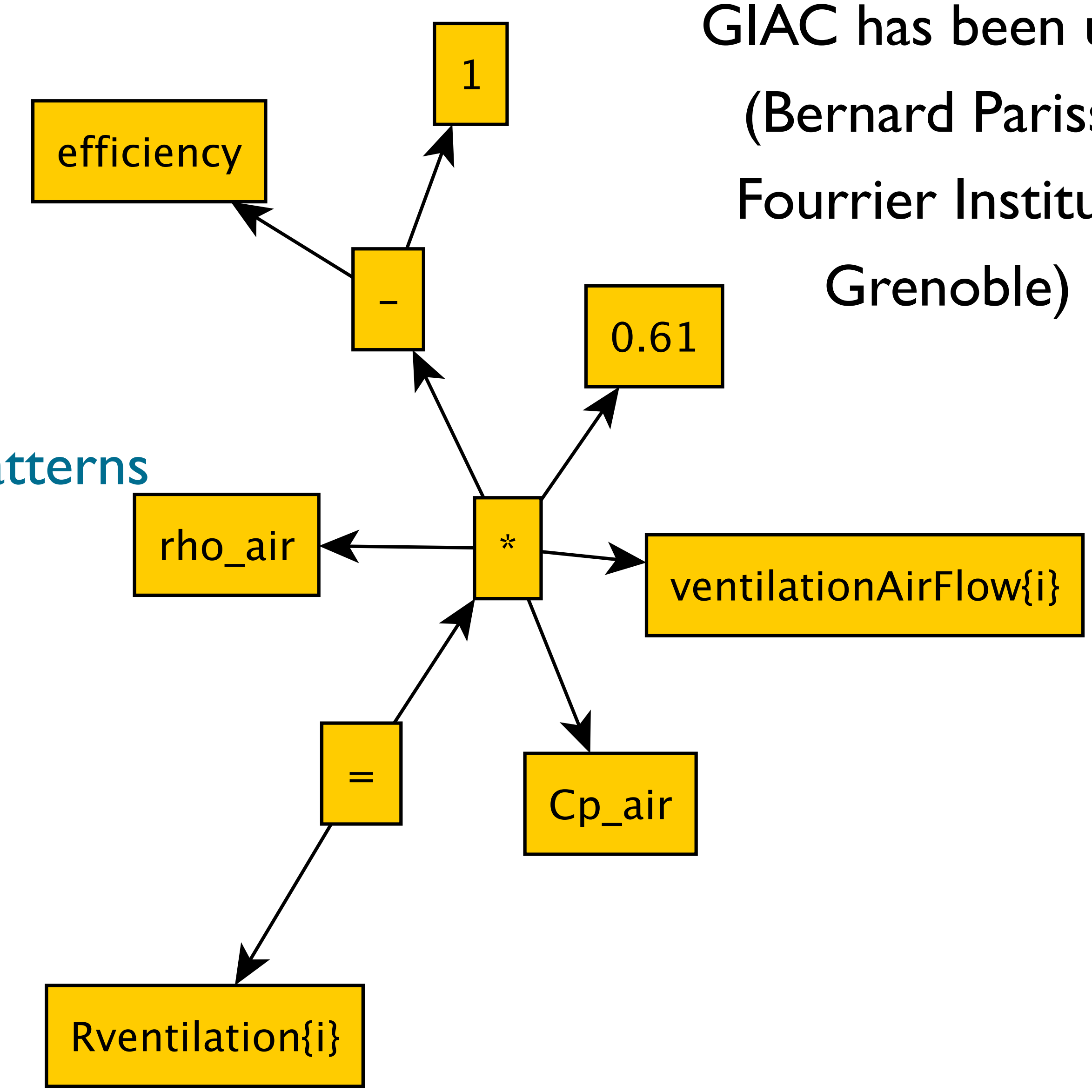
summary



symbolic manipulation:

models are represented by graphs

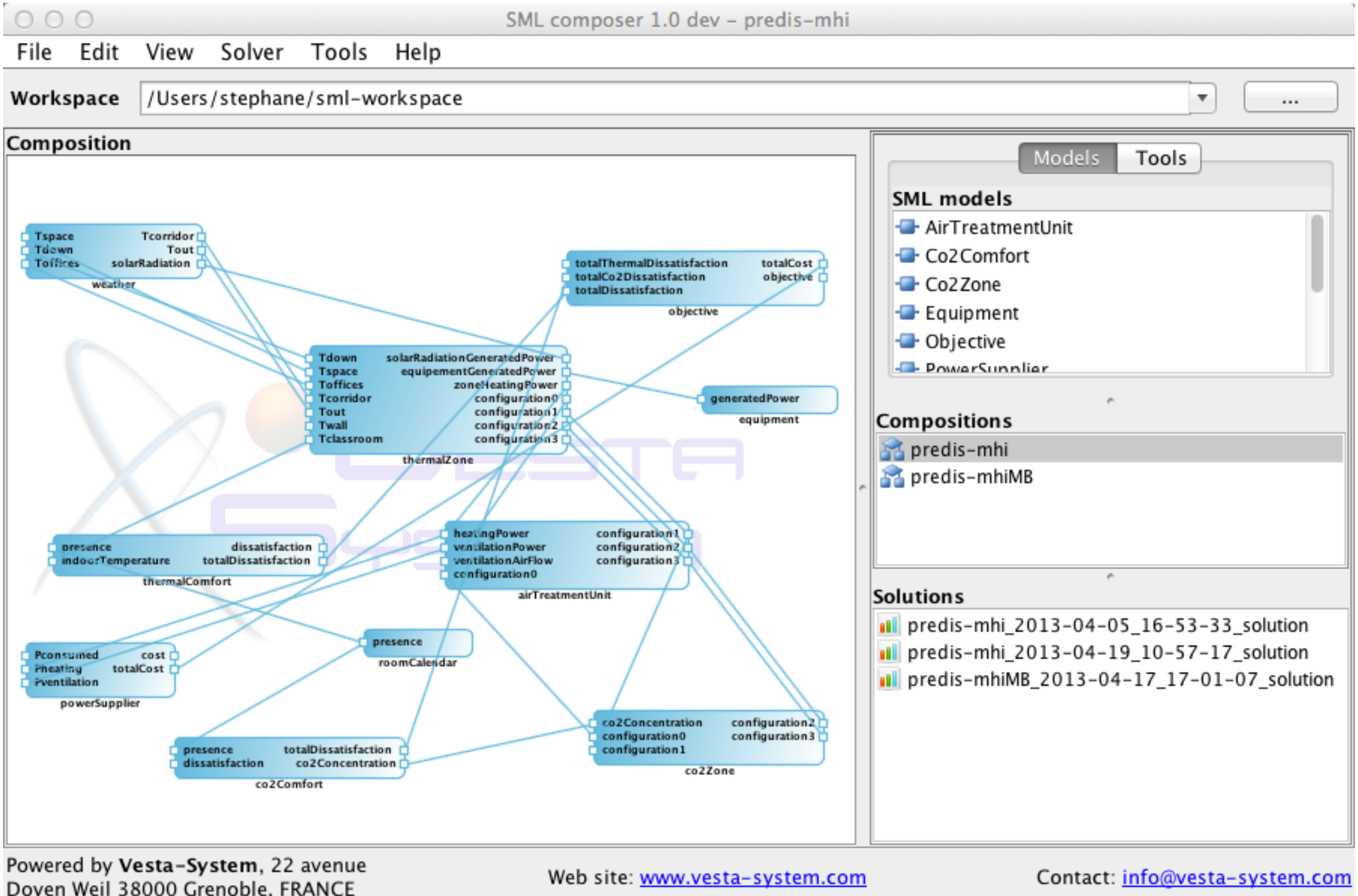
GIAC has been used (Bernard Parisse, Fourrier Institute, Grenoble)



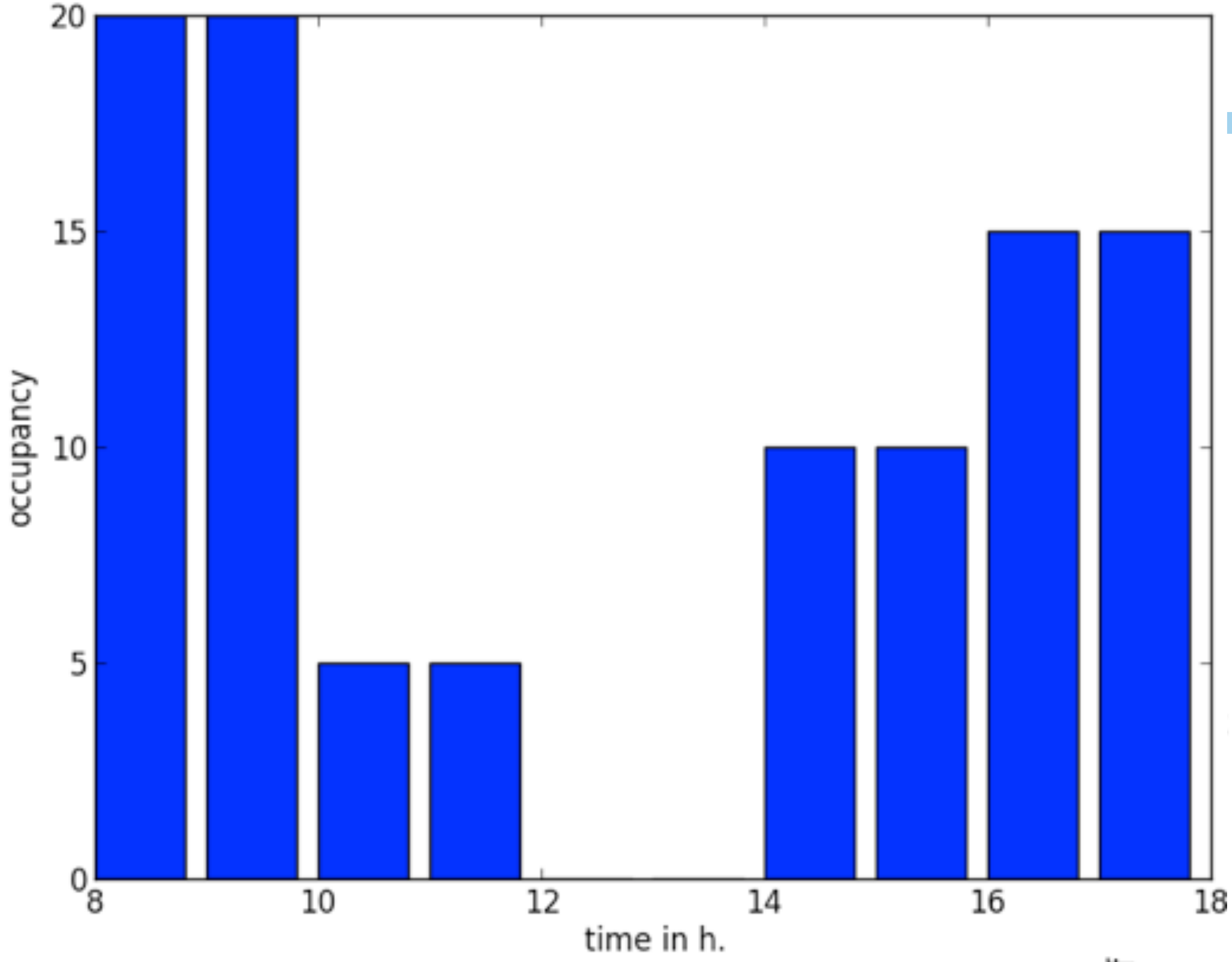

```
Rventilation = 1/((1-efficiency)*rho_air*Cp_air*0.61*ventilationAirFlow)
```

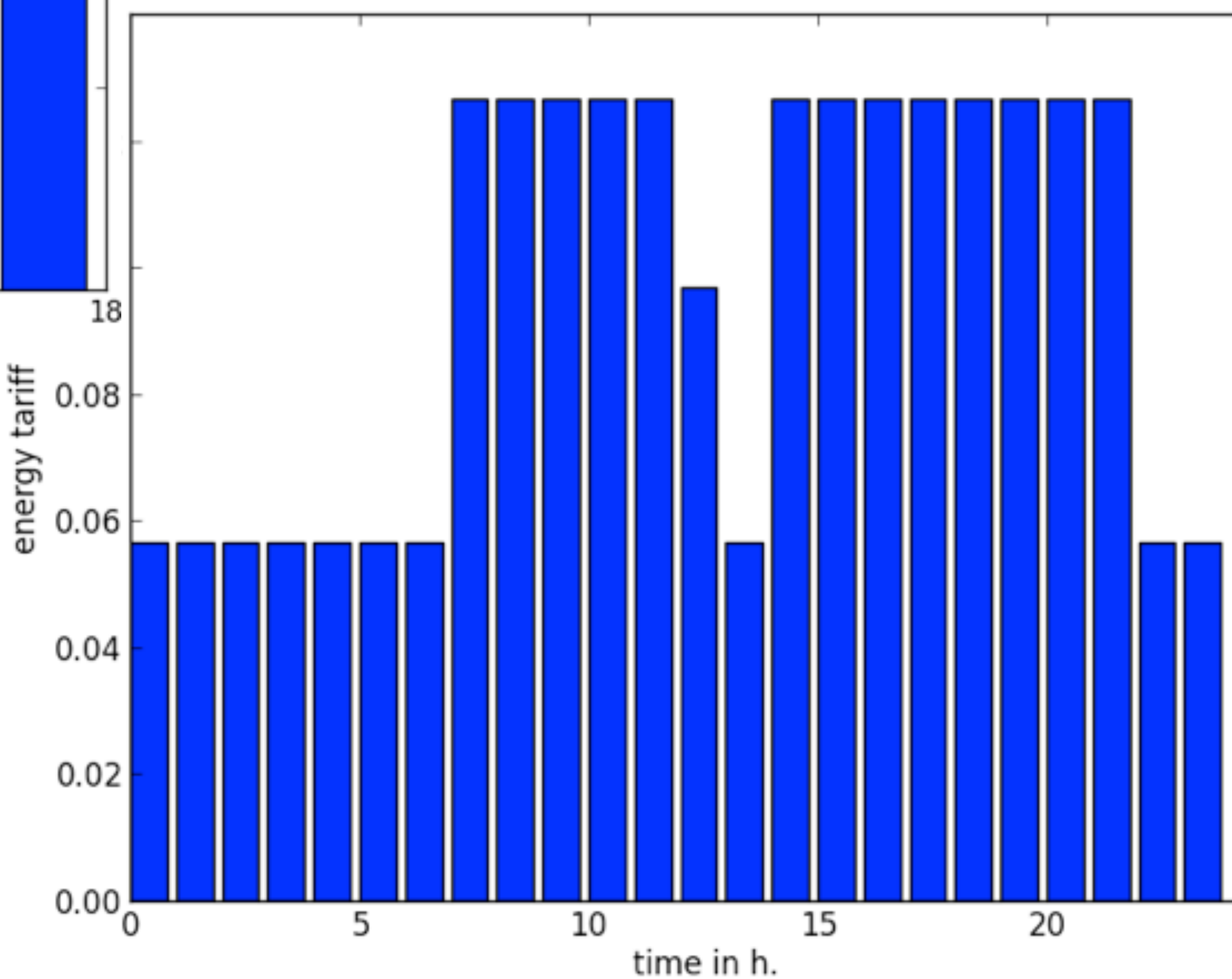
application example

a first version of composer exists...
... multi-application capabilities are coming

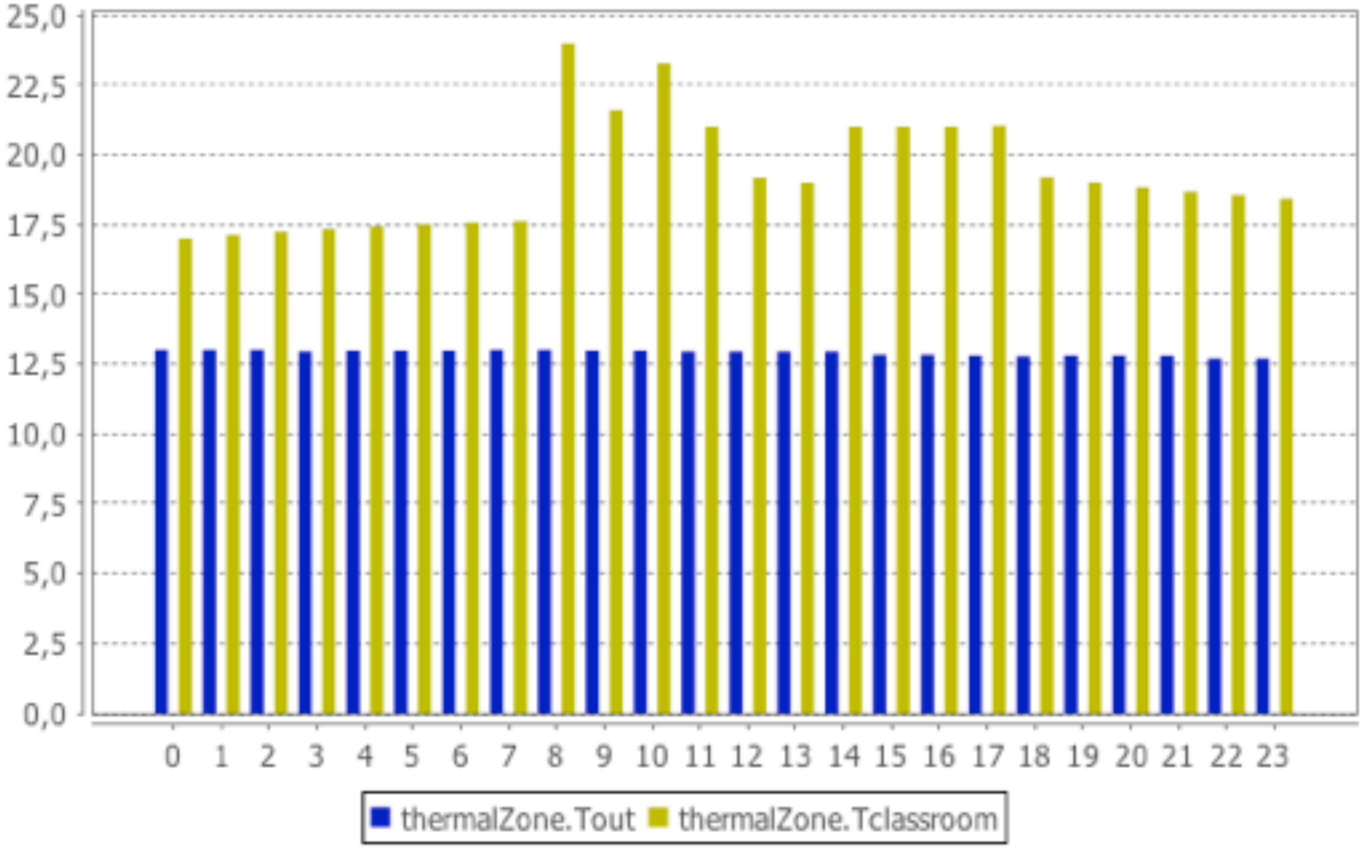


sample data

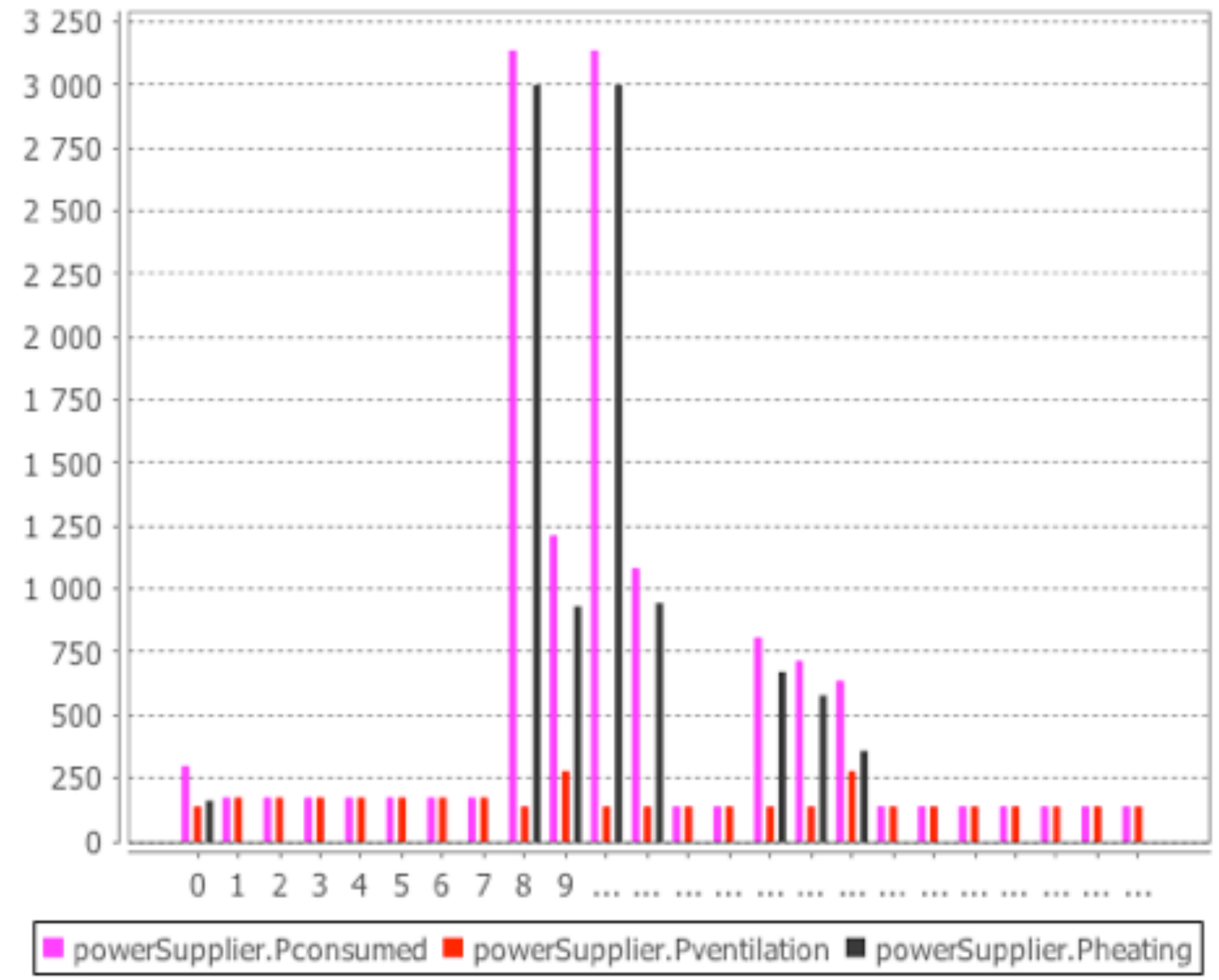

occupancy
0
5
10
15
20
8
10
12
14
16
18
time in h.


energy tariff
0.00
0.02
0.04
0.06
0.08
0
5
10
15
20
time in h.

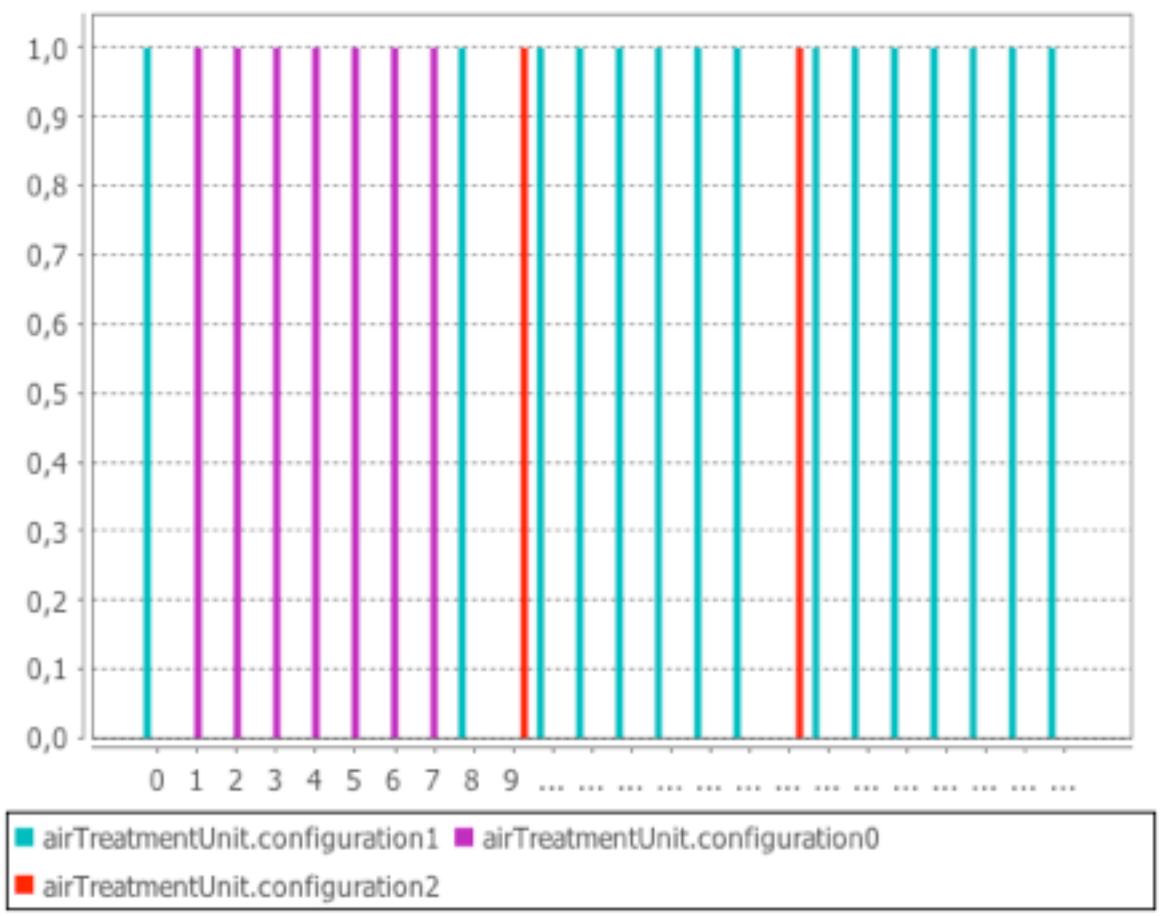
results



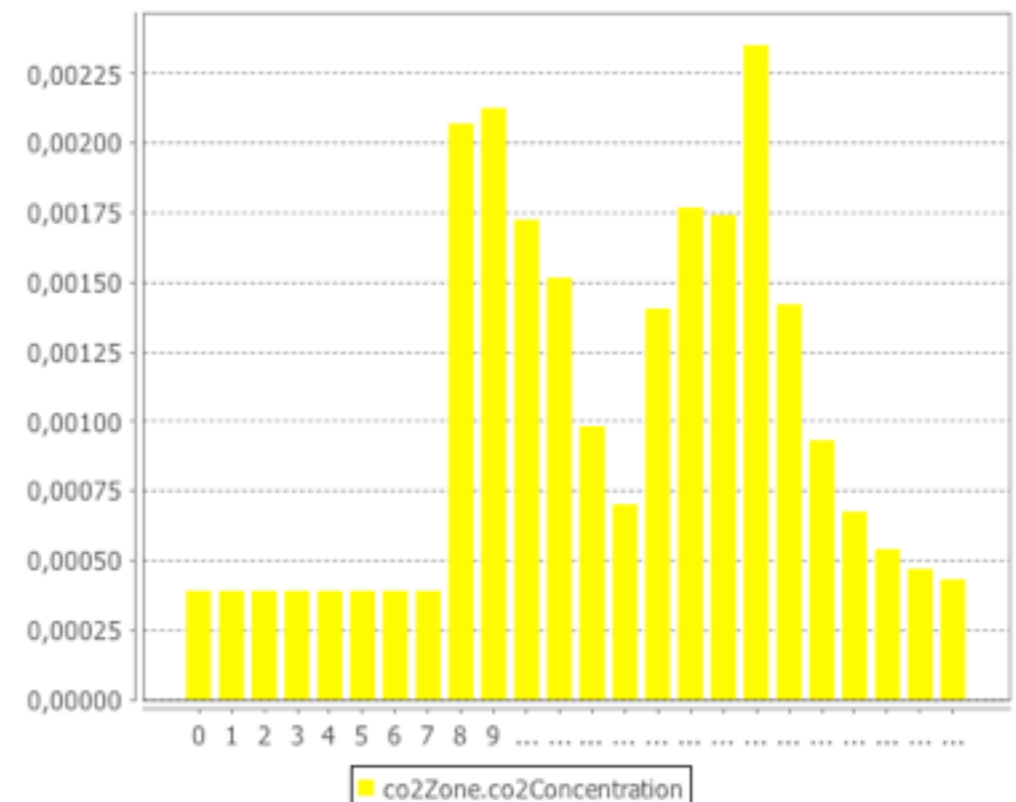
25,0
22,5
20,0
17,5
15,0
12,5
10,0
7,5
5,0
2,5
0,0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
thermalZone.Tout
thermalZone.Tclassroom



3 250
3 000
2 750
2 500
2 250
2 000
1 750
1 500
1 250
1 000
750
500
250
0
0 1 2 3 4 5 6 7 8 9
powerSupplier.Pconsumed
powerSupplier.Pventilation
powerSupplier.Pheating



1,0
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0,0
0 1 2 3 4 5 6 7 8 9
airTreatmentUnit.configuration1
airTreatmentUnit.configuration0
airTreatmentUnit.configuration2



0,00225
0,00200
0,00175
0,00150
0,00125
0,00100
0,00075
0,00050
0,00025
0,00000
0 1 2 3 4 5 6 7 8 9
co2Zone.co2Concentration

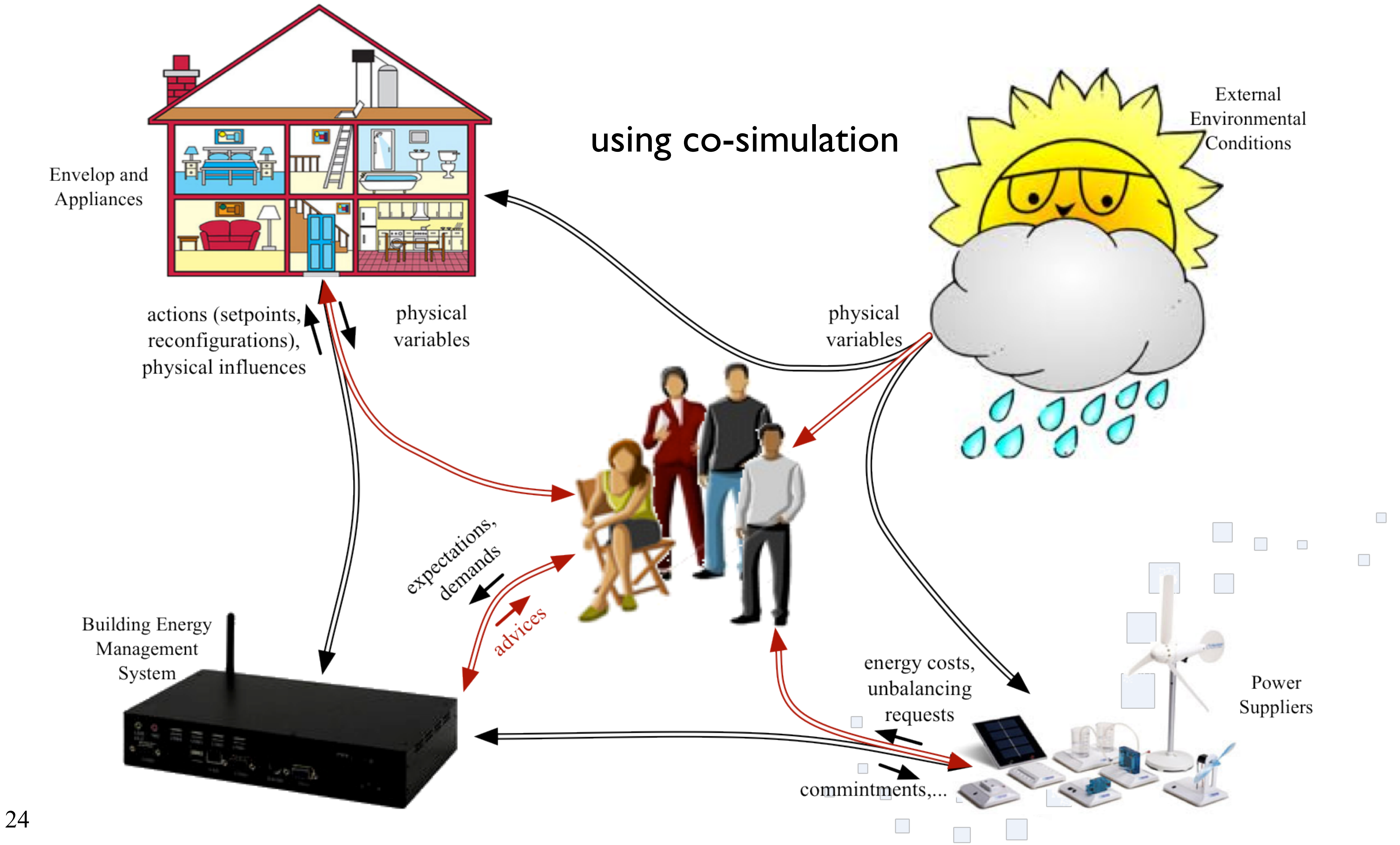
conclusions

- actually, two model processing systems have been developed: milp-workshop (research) and SMLcomposer (ergonomic)
- but both focus on anticipative energy management (and simulation)
  - formal manipulation of equations is still under development
- next applications:
  - parameter estimation (model learning)
  - diagnosis analysis
- remark: granularity cannot be changed by formal manipulation (other granularity = other neutral model)

modeling occupants

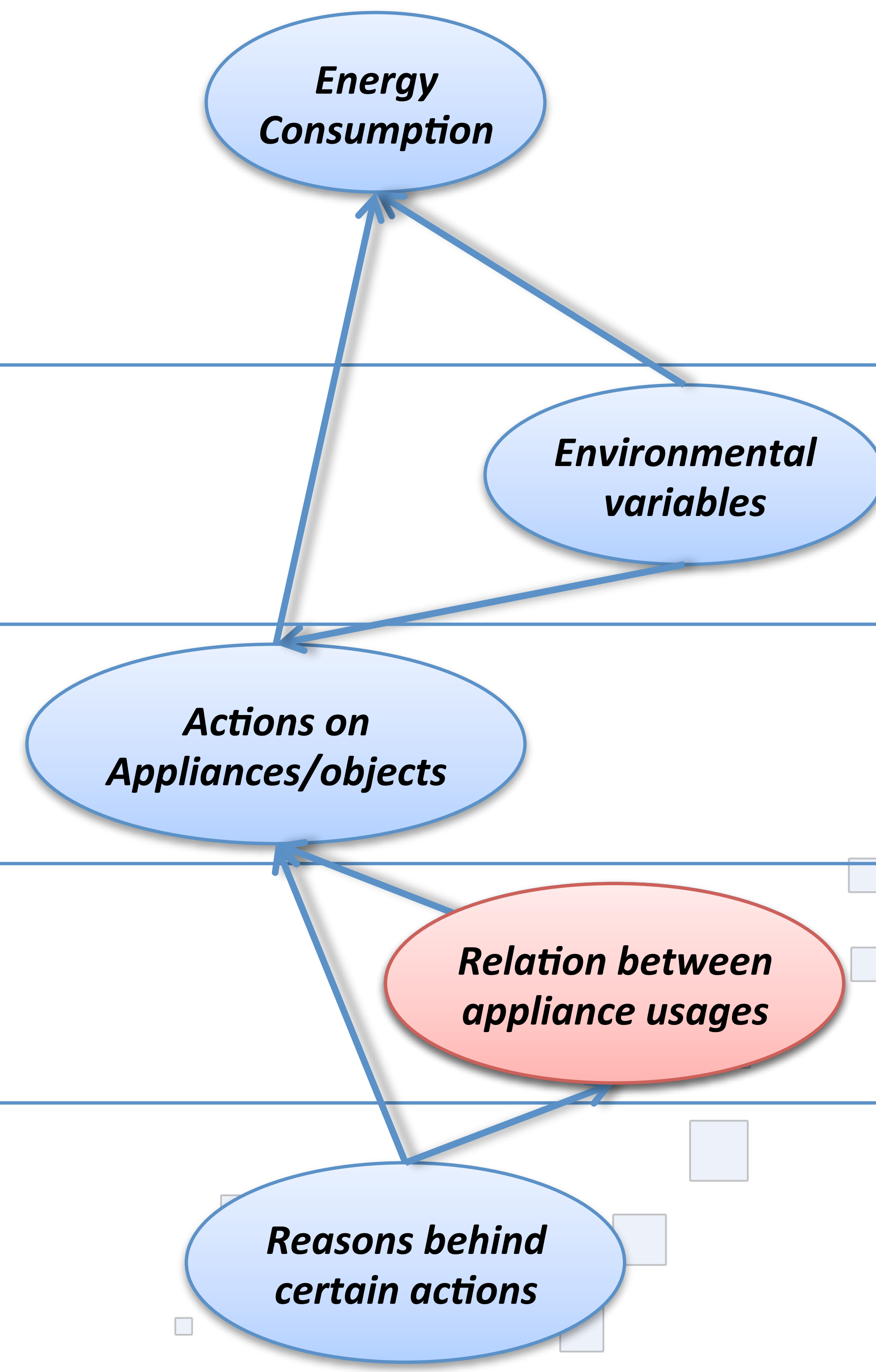
assessing the impact of energy management

using co-simulation

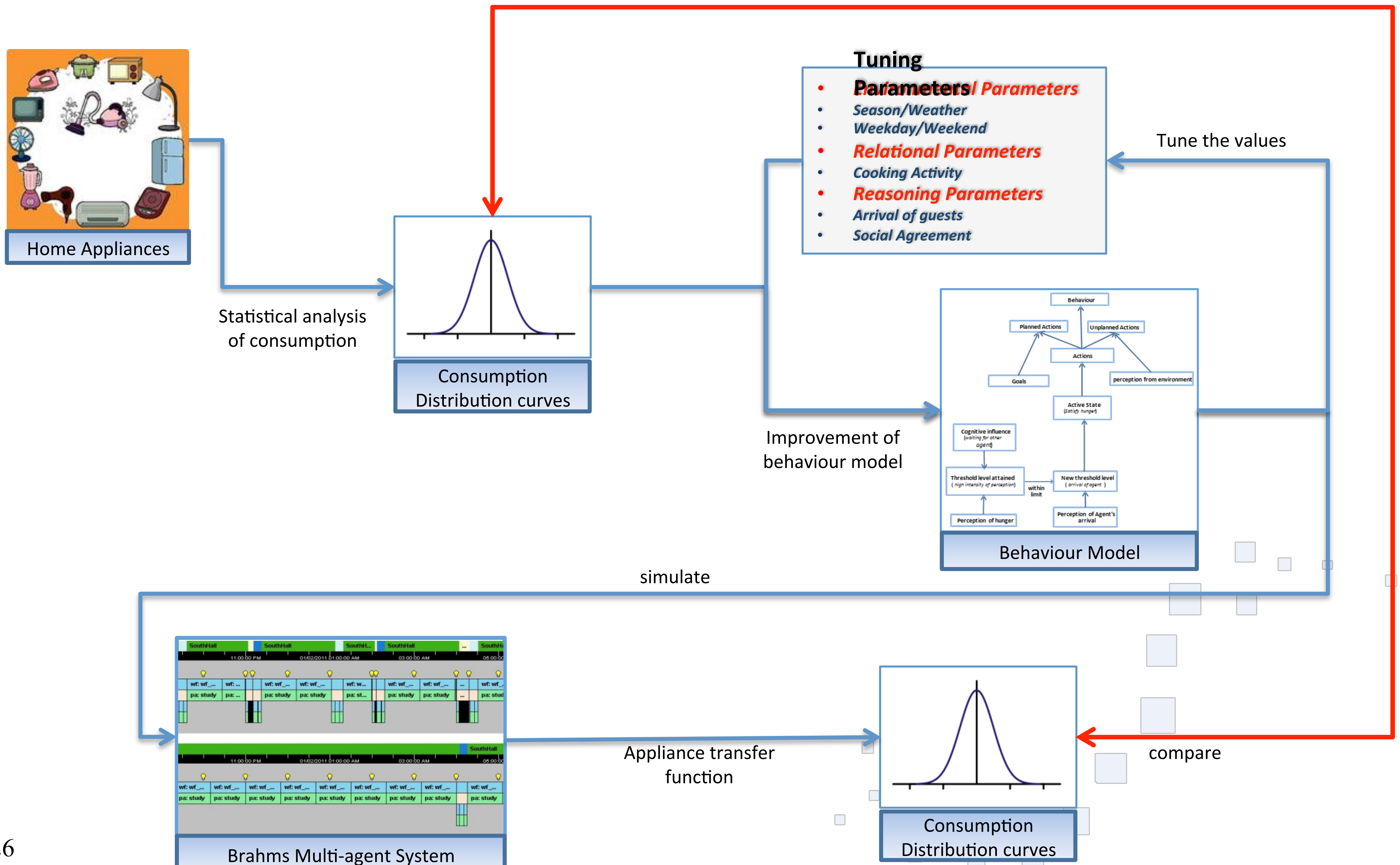


modeling process

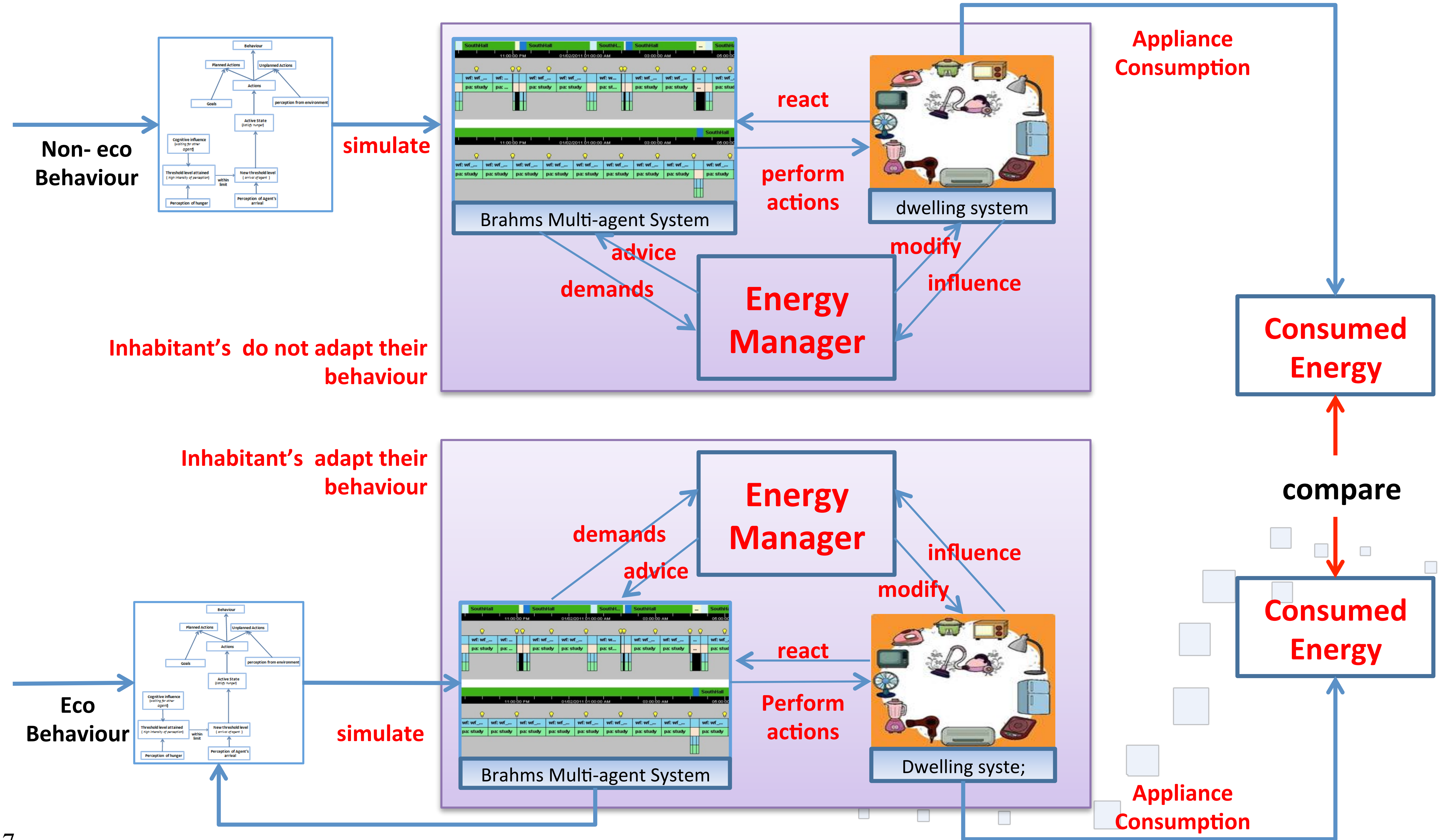
Local Behaviour (Personal experiments)	Global Behaviour (IRISE Database)	
	Weather, Season, Weekday, Weekend, Holiday, Day Time, presence of occupants, temperature etc…	**Level 1**
Perform specific actions e.g. open door of fridge, put food inside etc…….	*Turn on/off the appliance e.g. electric cooker…*	**Level 2**
	How the cooking activity effect the consumption of fridge etc……	**Level 3**
Why the cooker is used more on a particular day? Because there are guests at home….etc		**Level 4**



validation process

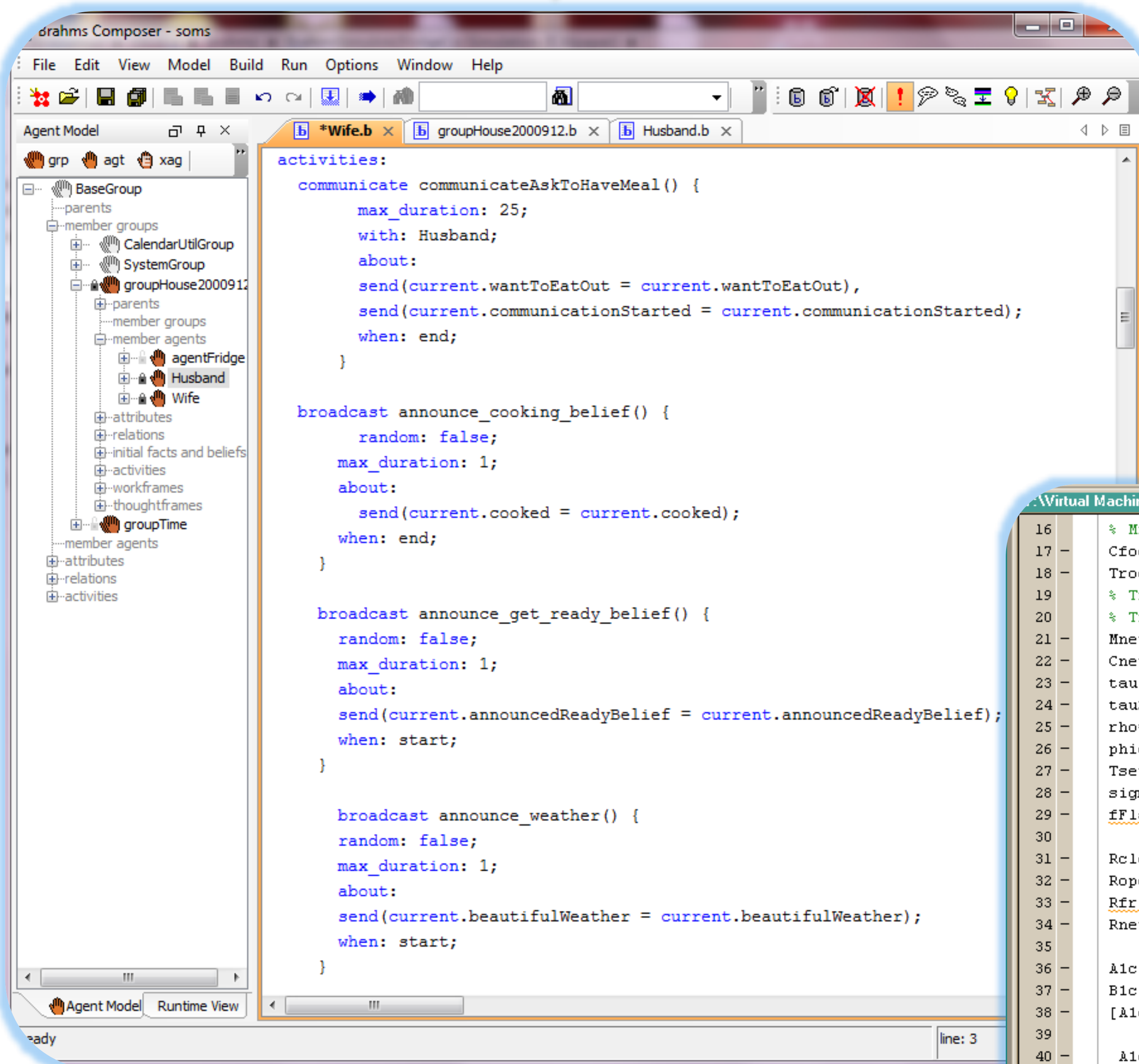


assessing the impact of energy management



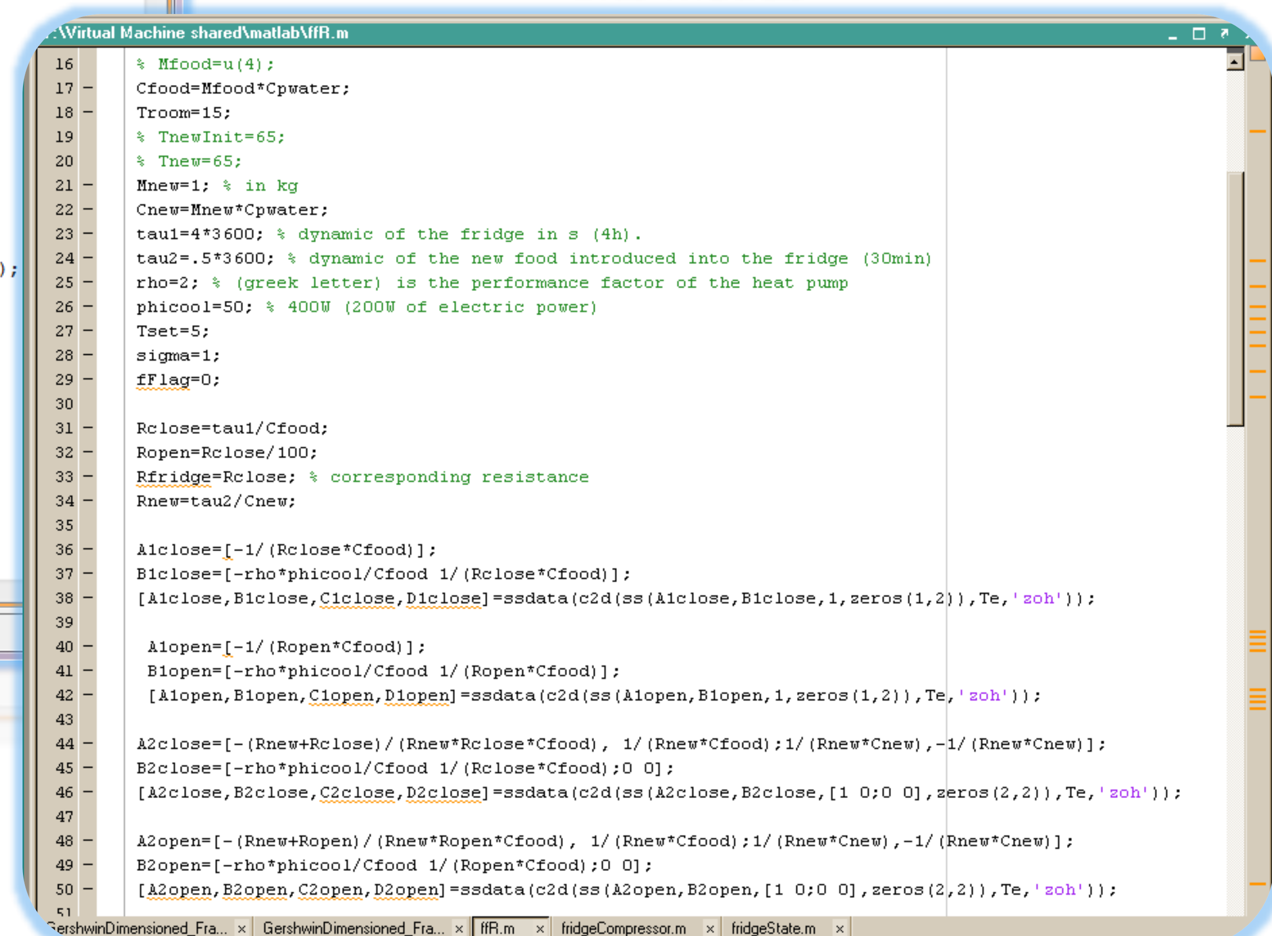
is there an unique global modeling language?

Behaviour model implemented into Brahms



> Believes, workframes, thoughtframes

Physical Model of refrigerator implemented into Matlab



?