Twomes: Digital Twins for the Home Heating Transition

Henri ter Hofte[1], Marco Winkelman[1], Hossein Rahmani[1] , Jeike Wallinga[1]
[1] Research Group Energy Transition, Windesheim University of Applied Sciences, Zwolle, the Netherlands
energietransitie@windesheim.nl

Which home heating model parameters of specific homes can we learn automatically from energy monitoring data in order to provide better advice to specific households about their home heating transition?

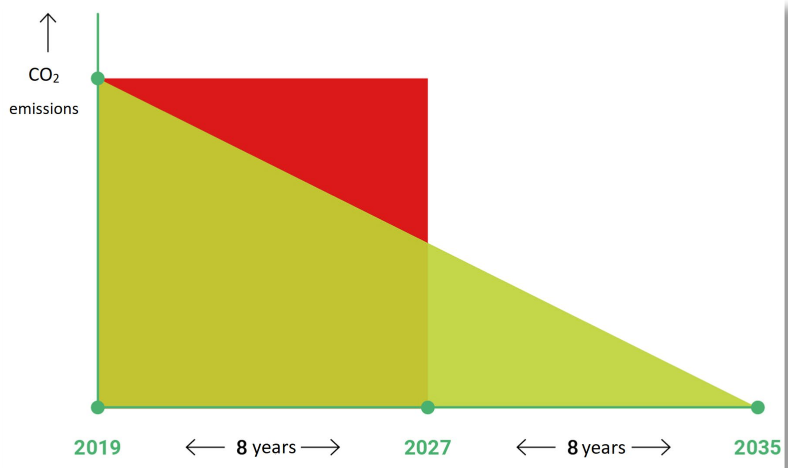
TECH FOR FUTURE
Centre of Expertise HTSM Oost

WHY?

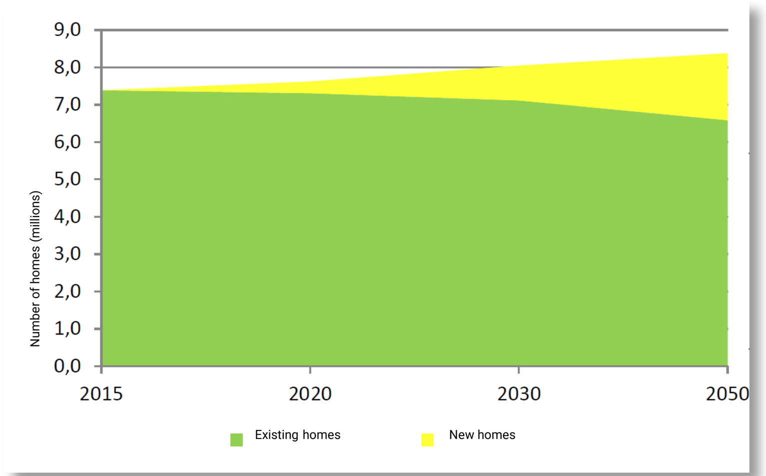
edu.nl/xghje

Scan for short Twomes movie

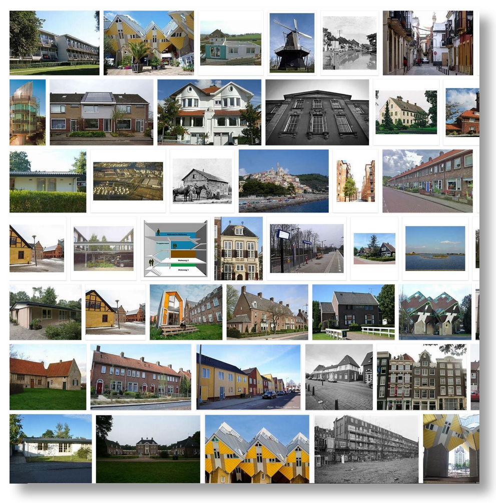
Paris agreement: limit temperature < 1.5 ˚C above pre-industrial levels.



Implies CO_2-emission budget; start reducing soon helps avoid cliff.



In NL, most homes in 2030 & 2050 were built before 2015.

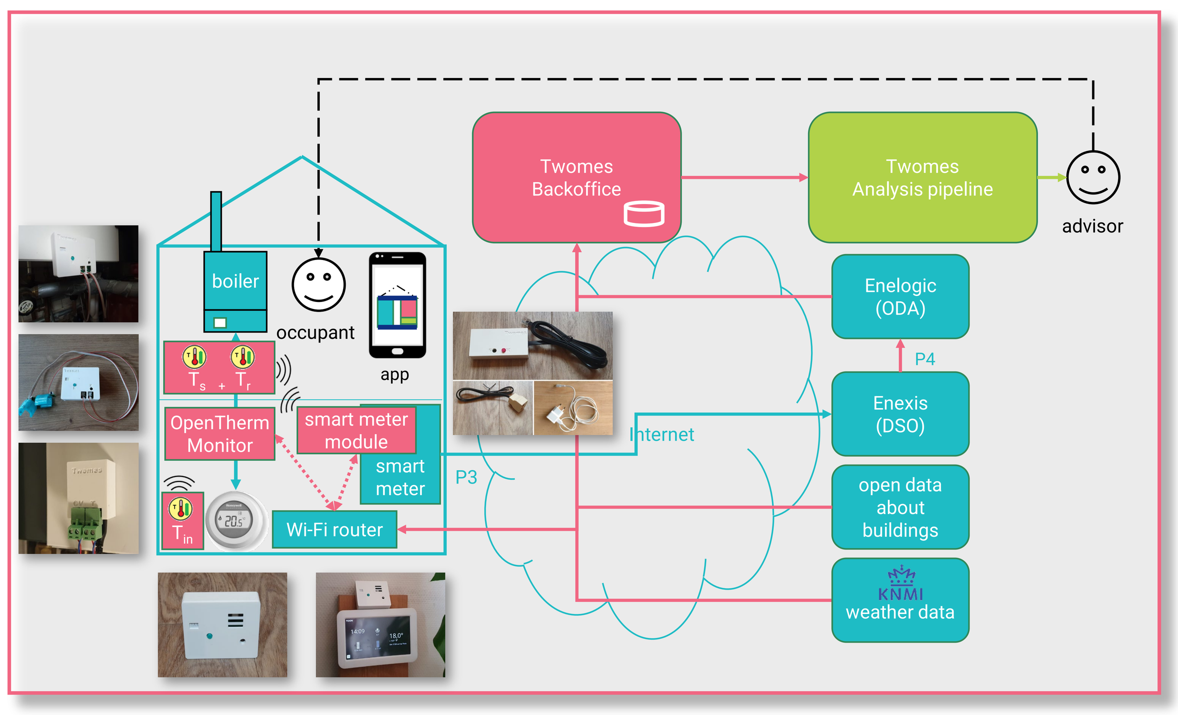


Homes and heating transition may vary.

DATA COLLECTION

edu.nl/demr9

Scan for open source software and hardware on GitHub



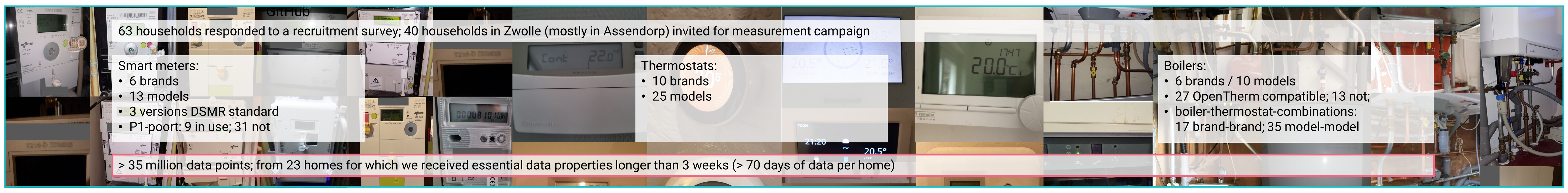
Measurement devices (cost including PCB, enclosure, power supply, cabling)							
Device	**Module**	**Cost**	**QR**	**Set A**	**Set B**	**Set C**	**Set D**
OpenTherm monitor	OpenTherm monitor	€ 25	✓	✓		✓	✓
smart meter module	smart meter module	€ 15	✓	✓			
smart meter module + boiler module + room monitor	smart meter module	€ 15	✓		✓	✓	✓
	boiler module	€ 25			✓	✓	✓
	room monitor	€ 20			✓		✓
	room monitor incl. CO_2-sensor	€ 50				✓	
Total per home				**€ 40**	**€ 60**	**€ 115**	**€ 85**

Data collected					
Category	**Measured data**	**Symbol**	**Unit**	**API**	**Sensor**
comfort	setpoint	T_{set}	°C		✓
weather	outdoor temperature	T_{out}	°C	KNMI	
	wind	U	m/s		
	global horizontal irradiation	I	W/m²		
indoor	indoor temperature	T_{in}	°C		✓
installation	supply temperature	T_s	°C		✓
	return temperature	T_r	°C		✓
heating energy	electricity	E	kWh	Enelogic	✓
	gas	G	m³		
occupancy/ ventilation	CO_2 concentration	CO_2	ppm		✓
	Bluetooth presence	BT_{pres}	#pp		✓

SUBJECTS & DATA

edu.nl/aqnhv

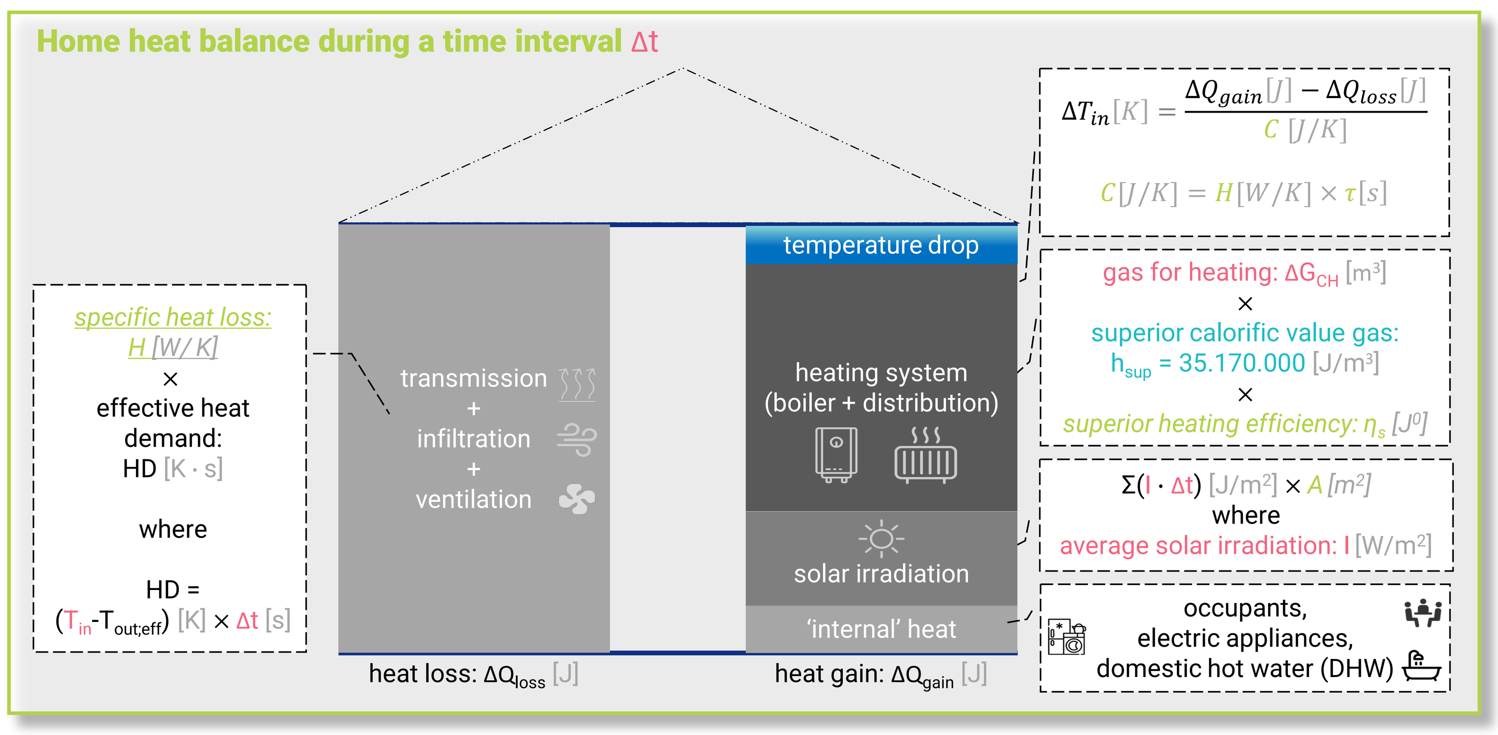
Scan for dataset

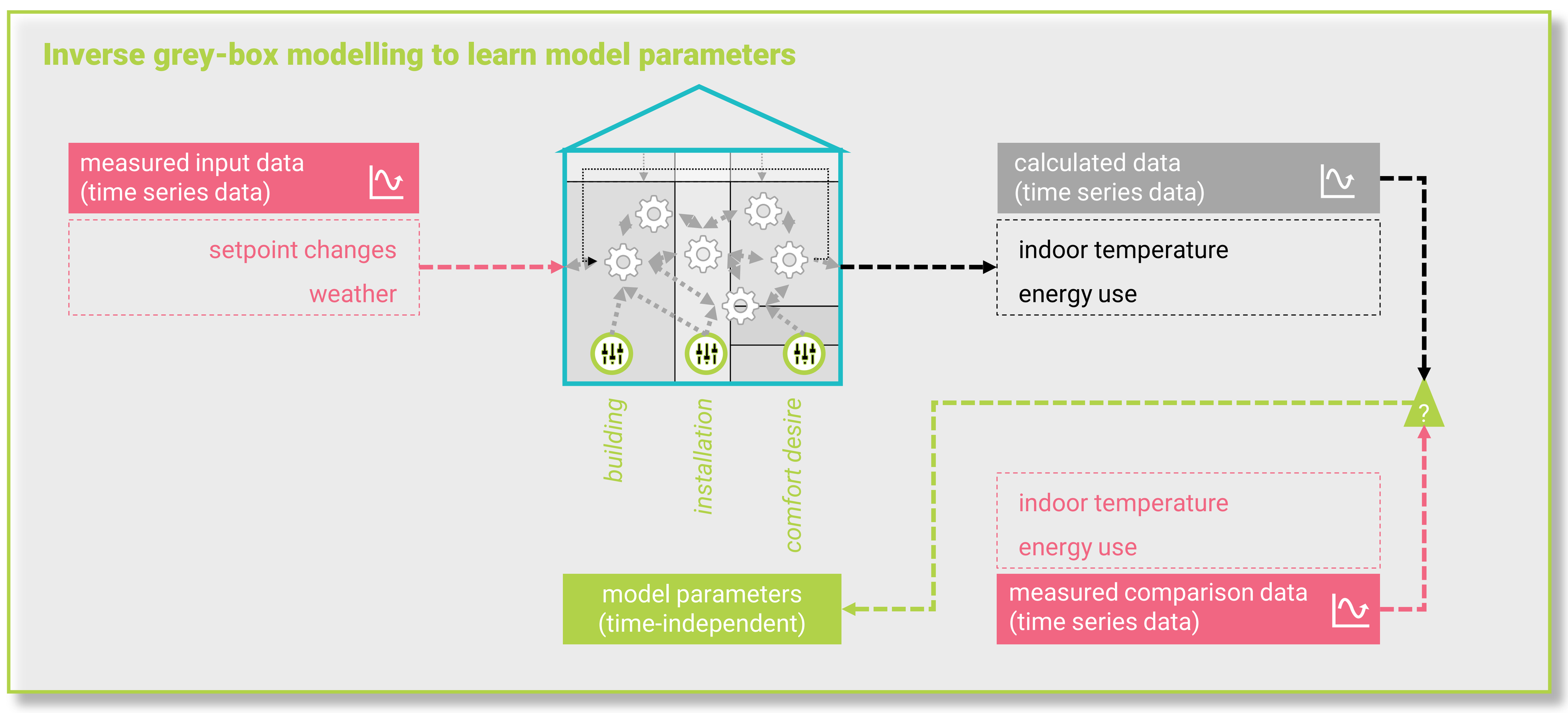


DATA ANALYSIS

edu.nl/kttbe

Scan for GEKKO Python heat balance model






```
##########################################
# Gekko Model - Initialize
##########################################

# initialize gekko
m = GEKKO(remote=False)
m.time = np.arange(0, duration_s, step_s)
```

```
##########################################################
# Gekko Model - Model Parameters
##########################################################

# Model parameter: H [W/K]: specific heat loss
H_W_p_K = m.FV(value=300.0, lb=0, ub=1000)
H_W_p_K.STATUS = 1; H_W_p_K.FSTATUS = 0

# Model parameter: tau [s]: effective thermal inertia
tau_s = m.FV(value=(100 * s_p_h), lb=(10 * s_p_h), ub=(1000 * s_p_h))
tau_s.STATUS = 1; tau_s.FSTATUS = 0
```

```
##########################################################################
# Equation - Q_gain_sol_avg_W: : heat gain from solar irradiation
##########################################################################

# Model parameter/fixed value: A [m^2]: Effective area of the solar aperture
if np.isnan(iterator_A_m2):
    A_m2 = m.FV(value=5, lb=1, ub=100); A_m2.STATUS = 1; A_m2.FSTATUS = 0
else:
    A_m2 = m.Param(value=iterator_A_m2)

irradiation_hor_avg_W_p_m2 = m.MV(value=irradiation_hor_avg_W_p_m2_array)
irradiation_hor_avg_W_p_m2.STATUS = 0; irradiation_hor_avg_W_p_m2.FSTATUS = 1

Q_gain_sol_avg_W = m.Intermediate(irradiation_hor_avg_W_p_m2 * A_m2)
```

```
##########################################################################
# Main Equations & Solver
##########################################################################
Q_gain_W = m.Intermediate(Q_gain_gas_CH_avg_W + Q_gain_sol_avg_W + Q_gain_int_avg_W)
Q_loss_W = m.Intermediate(H_W_p_K * (T_in_avg_C - T_out_e_avg_C))
C_J_p_K  = m.Intermediate(H_W_p_K * tau_s)
m.Equation(T_in_avg_C.dt() == ((Q_gain_W - Q_loss_W) / C_J_p_K))

m.options.IMODE = 5
m.options.EV_TYPE = ev_type # specific objective function (L1-norm vs L2-norm)
m.solve(False)
```

RESULTS & CONCLUSIONS

edu.nl/jw9j3

Scan for more results

Model parameters to learn			
symbol	scope	parameter	unit
H	building	specific heat loss	W/K
τ	building	thermal inertia	s (h)
C	building	thermal mass (C = H × τ)	J/K (Wh/K)
A	building	apparent horizontal window area	m²
P	installation	maximum heating system power	W
η_s	installation	superior heating system efficiency	J⁰
CD	behaviour	comfort desire (thermostat setpoints)	K·s

Initial results

Building parameters can be learned:

- specific heat loss: **H** [W/K],
- thermal mass: **C** [W/K] (or [Wh/K])
- thermal inertia: **τ** [s] (or [h])

What was challenging

- outlier removal (in particular for smart meter data)
- interpolation (in particular for smart meter timestamps)
- GEKKO Python model (validated with virtual home data)
- 10 to 50-fold increased analysis speed after switch to RMSE (i.e., using EV_TYPE=2, instead of EV_TYPE=1)

What's next

- assess increased precision over calculating parameters based on public building data
- learn installation parameters
- learn infiltration and ventilation parameters
- assess utility for occupant and advisor
- use to assess real effect of interventions