

Twomes: Digital Twins for the Home Heating Transition

Henri ter Hofte¹, Marco Winkelman¹, Hossein Rahmani¹, Jeike Wallinga¹
¹ Research Group Energy Transition, Windesheim University of Applied Sciences, Zwolle, the Netherlands

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?



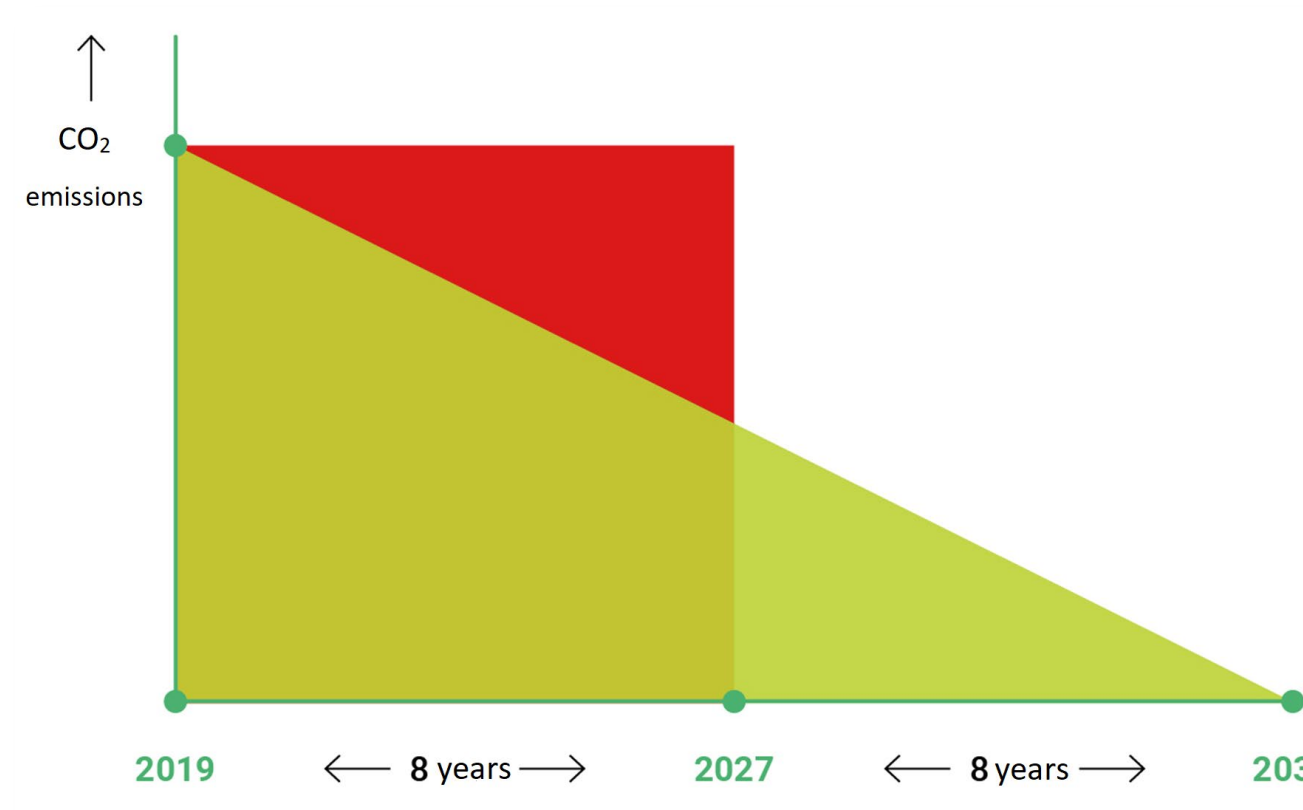
WHY?



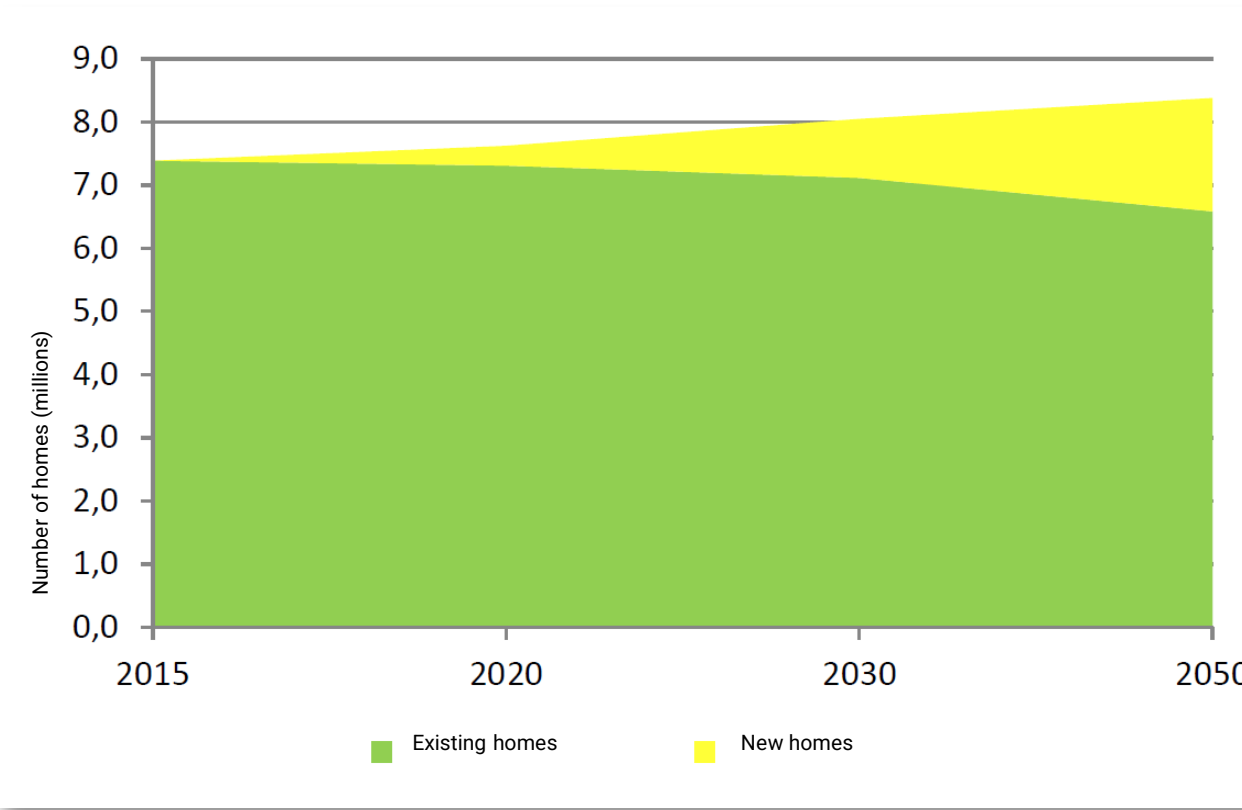
Scan for short Twomes movie



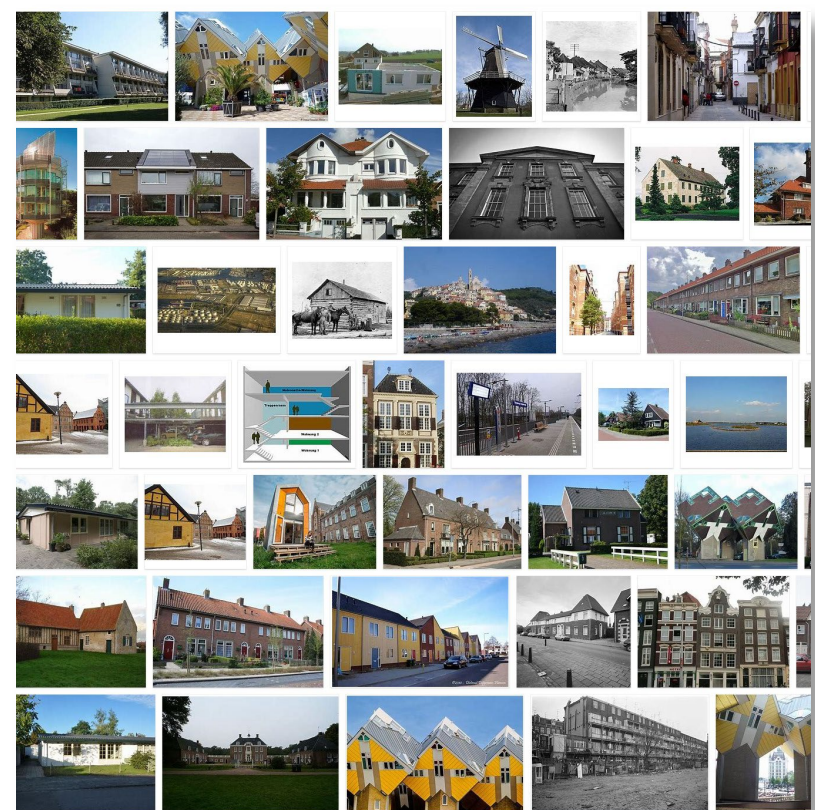
Paris agreement: limit temperature < 1.5 °C above pre-industrial levels.



Implies CO₂-emission budget; start reducing soon helps avoid cliff.

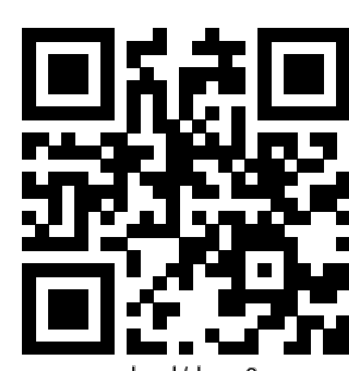


In NL, most homes in 2030 & 2050 were built before 2015.

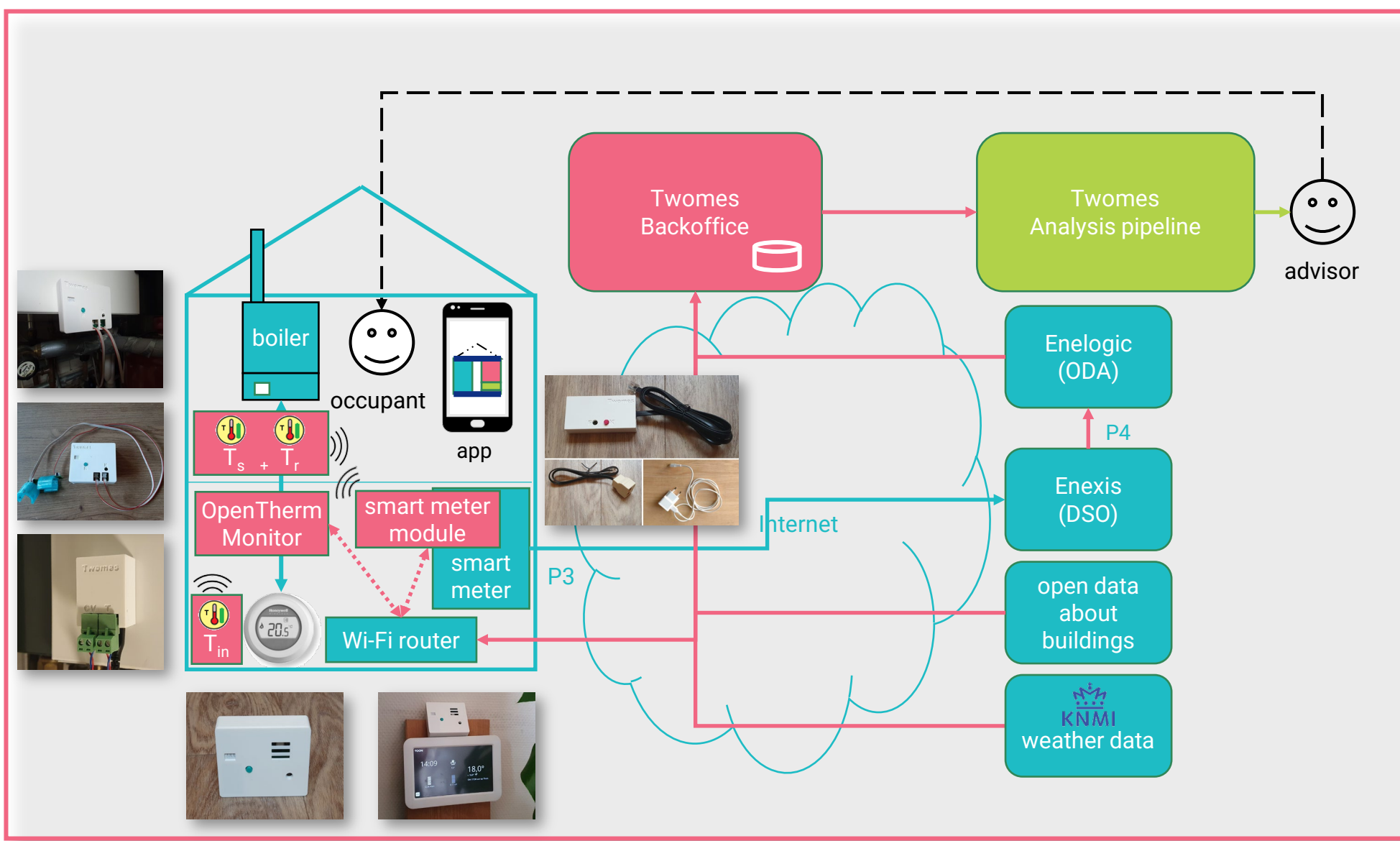


Homes and heating transition may vary.

DATA COLLECTION



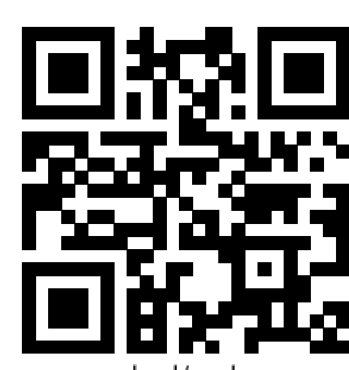
Scan for open source software and hardware on GitHub



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		✓
	outdoor temperature	T _{out}	°C		
weather	wind	U	m/s	KNMI	
	global horizontal irradiation	I	W/m ²		
indoor	indoor temperature	T _{in}	°C		✓
installation	supply temperature	T _s	°C		✓
	return temperature	T _r	°C		✓
heating energy	electricity	E	kWh	Enellogic	✓
	gas	G	m ³		
occupancy/ventilation	CO ₂ concentration	CO ₂	ppm		✓
	Bluetooth presence	BT _{pres}	#pp		✓

SUBJECTS & DATA



Scan for dataset

63 households responded to a recruitment survey; 40 households in Zwolle (mostly in Assendorp) invited for measurement campaign

Smart meters:

- 6 brands
- 13 models
- 3 versions DSMR standard
- P1-poort: 9 in use; 31 not

Thermostats:

- 10 brands
- 25 models

Boilers:

- 6 brands / 10 models
- 27 OpenTherm compatible; 13 not;
- boiler-thermostat-combinations: 17 brand-brand; 35 model-model

> 35 million data points; from 23 homes for which we received essential data properties longer than 3 weeks (> 70 days of data per home)



DATA ANALYSIS



Scan for GEKKO Python heat balance model

Home heat balance during a time interval Δt

specific heat loss: $H [W/K]$ × effective heat demand: $HD [K \cdot s]$

where $HD = (T_{in} - T_{out,eff}) [K] \times \Delta t [s]$

transmission + infiltration + ventilation

heating system (boiler + distribution)

solar irradiation

'internal' heat

heat loss: $\Delta Q_{loss} [J]$

heat gain: $\Delta Q_{gain} [J]$

gas for heating: $\Delta Q_{CH} [m^3]$ × superior calorific value gas: $\dot{h}_{sup} = 35.170.000 [J/m^3]$ × superior heating efficiency: $\eta_s [J^0]$

$\Delta T_{in} [K] = \frac{\Delta Q_{gain} [J] - \Delta Q_{loss} [J]}{C [J/K]}$

$C [J/K] = H [W/K] \times \tau [s]$

average solar irradiation: $I [W/m^2]$

where $\Sigma(I \cdot \Delta t) [J/m^2] \times A [m^2]$

occupants, electric appliances, domestic hot water (DHW)

Inverse grey-box modelling to learn model parameters

measured input data (time series data)

setpoint changes

weather

building

installation

comfort desire

calculated data (time series data)

indoor temperature

energy use

indoor temperature

energy use

measured comparison data (time series data)

model parameters (time-independent)

```
# 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

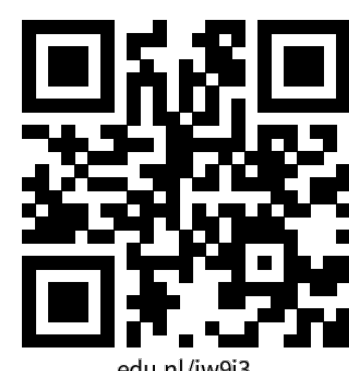
# Equation - Q_gain_sol_avg_W: heat gain from solar irradiation
if np.isnan(iterator_A_m2):
    A_m2 = m.FV(value=5, lb=1, ub=100); A_m2.STATUS = 1; A_m2.FSTATUS = 0
else:
    A_m2 = m.Param(value=iterator_A_m2)

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



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	m ²
P	installation	maximum heating system power	W
η_s	installation	superior heating system efficiency	J ⁰
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

