



Universiteit Utrecht



EVCONSULT  
experts in sustainable mobility



# The ABCD model

How to more accurately predict  
and manage the energy transition



Provincie Noord-Brabant



Technische Universiteit  
Eindhoven  
University of Technology

Where innovation starts



# Digging and burning is sooo 2<sup>nd</sup> millennium

## We are in the middle of the 5<sup>th</sup> energy revolution

**1** Fire & language 150W pp 0,15GW total



**2** Agriculture & writing 500W pp 15GW total



**3** Coal & printing 4000W pp 500GW total

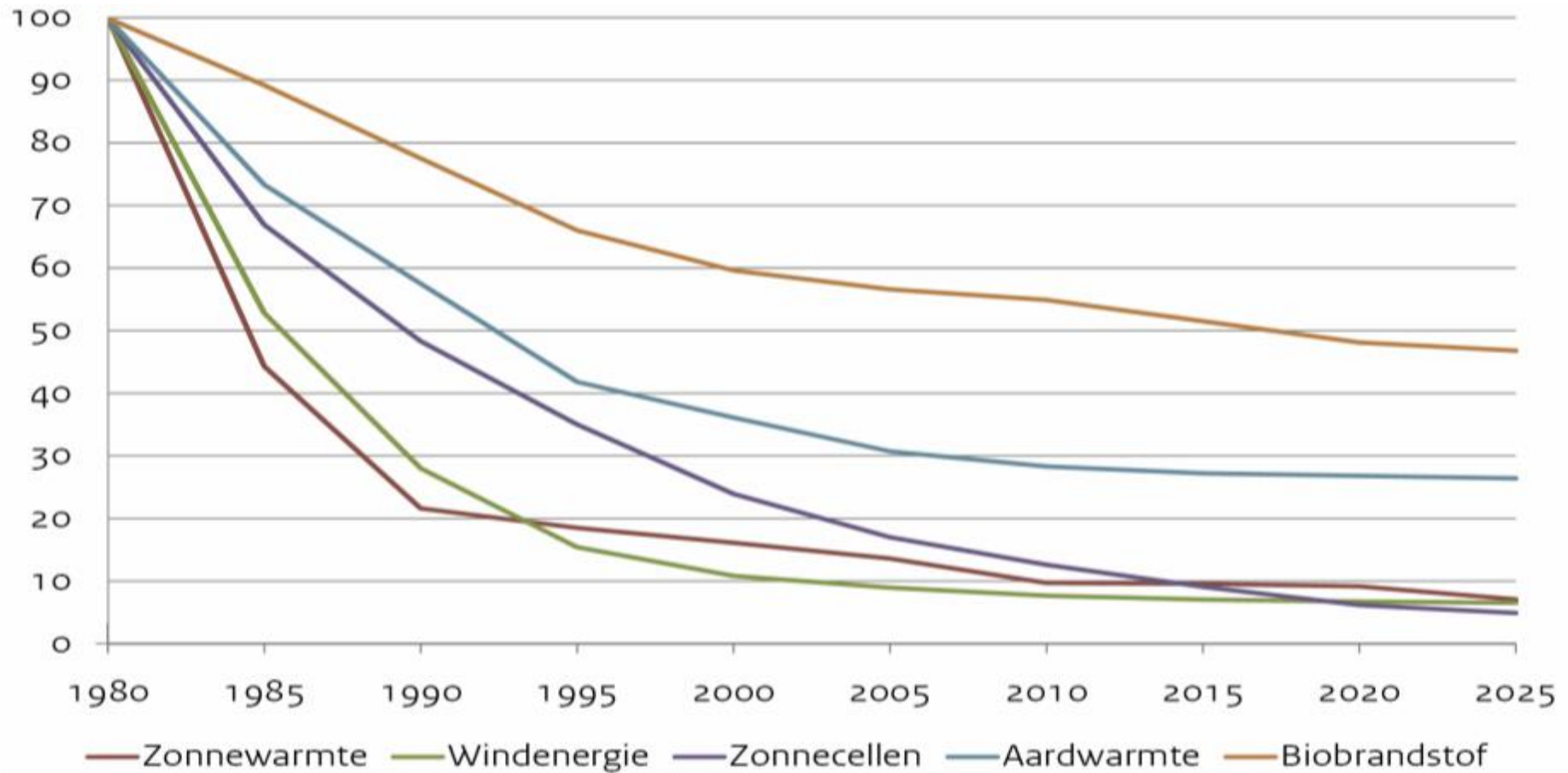


**4** Oil & telecom 11000W pp 15 TW total



# Renewables trade raw materials for knowledge

## Good for the environment but also cheaper



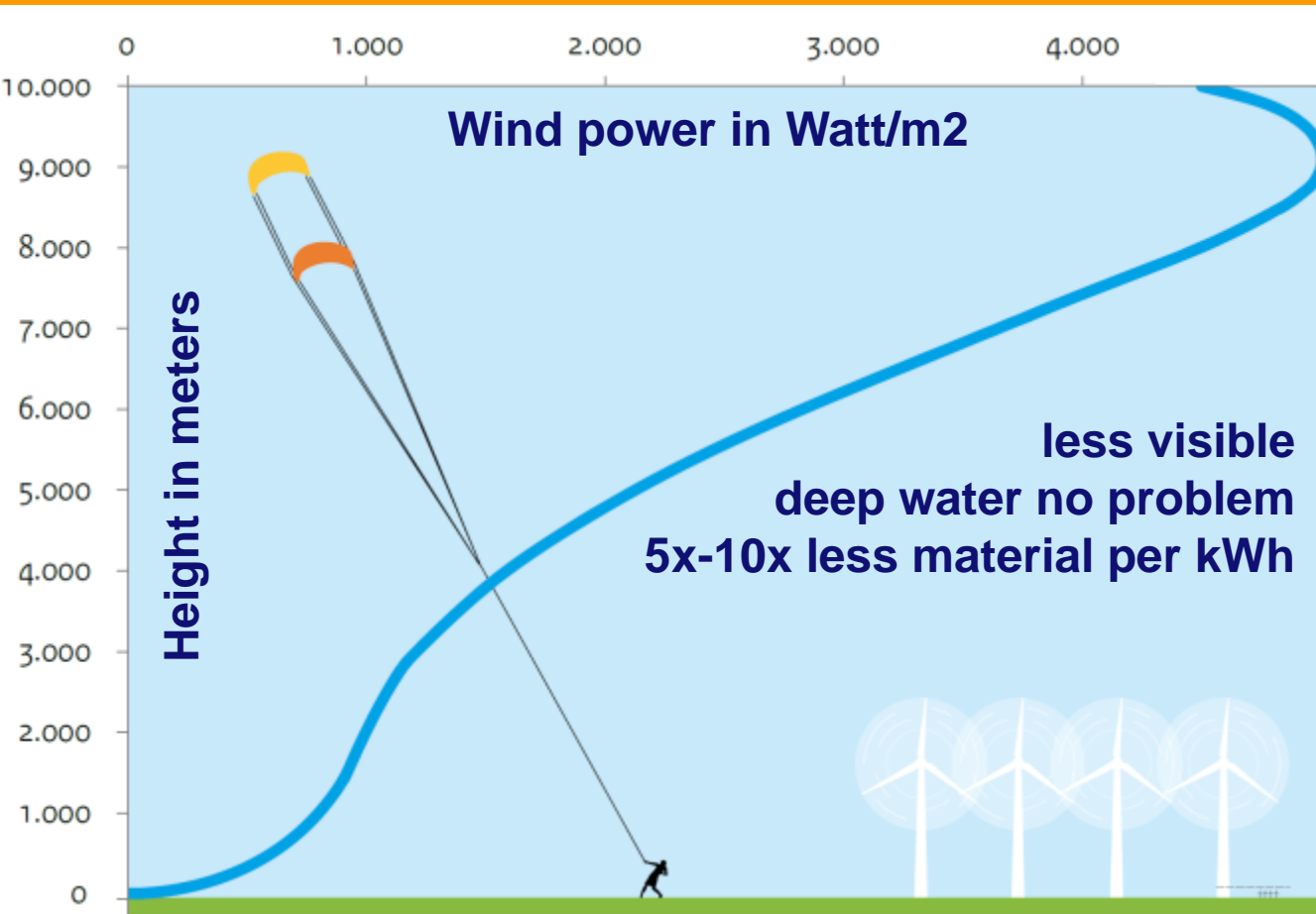
# Wind is becoming cheaper

## Offshore already 5.5 cents/kWh





# It will become much cheaper still Especially with airborne wind energy (AWE)



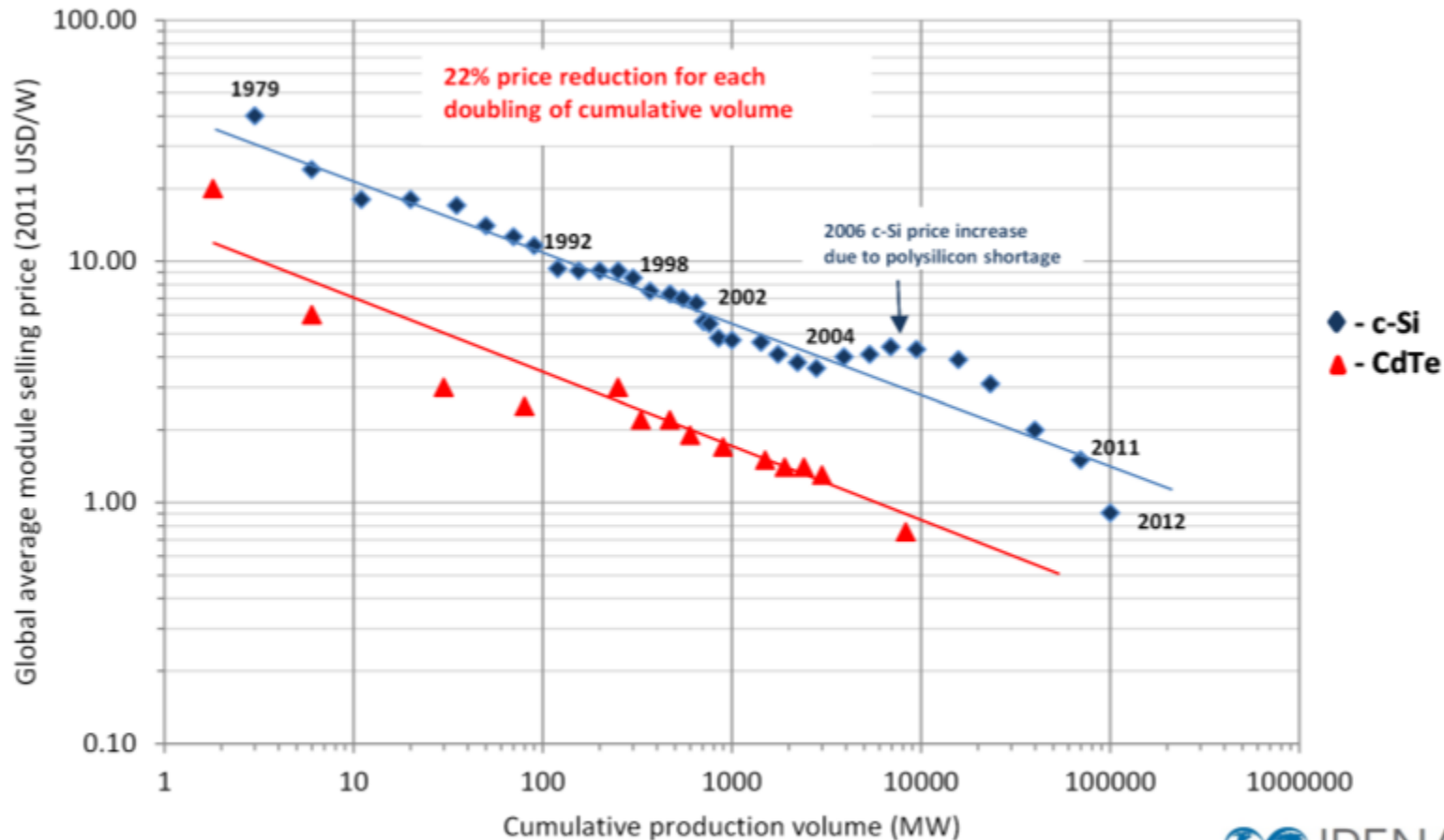
# Room enough on the North Sea

Red is windmills, blue is floatovoltaics



# Solar is becoming cheaper

## 100x cheaper since 1980s... Imagine that for oil...



If you look at raw material cost, solar could become 1 or 2 cents per kWh



AukeHoekstra @AukeHoekstra · Aug 27

In a desert but still: less than \$0.03/kWh!  
Solar Delivers Cheapest Electricity 'Ever,  
Anywhere, By Any Technology'





# My roof is a money maker



**Now let's look at EVs**  
**Did you know the first racecar was electric?**





Did you know the electric motor is...  
3x more efficient, 3x lighter and 30x smaller?





**Did you know the battery weight  
already decreased 20x since 1900?**

**Did you know the fastest accelerating (0-100 in 2.3 sec) production car is an electric family car?**



**Did you know that for every turn of the blades of one large windmill, an EV is propelled about 10km?**



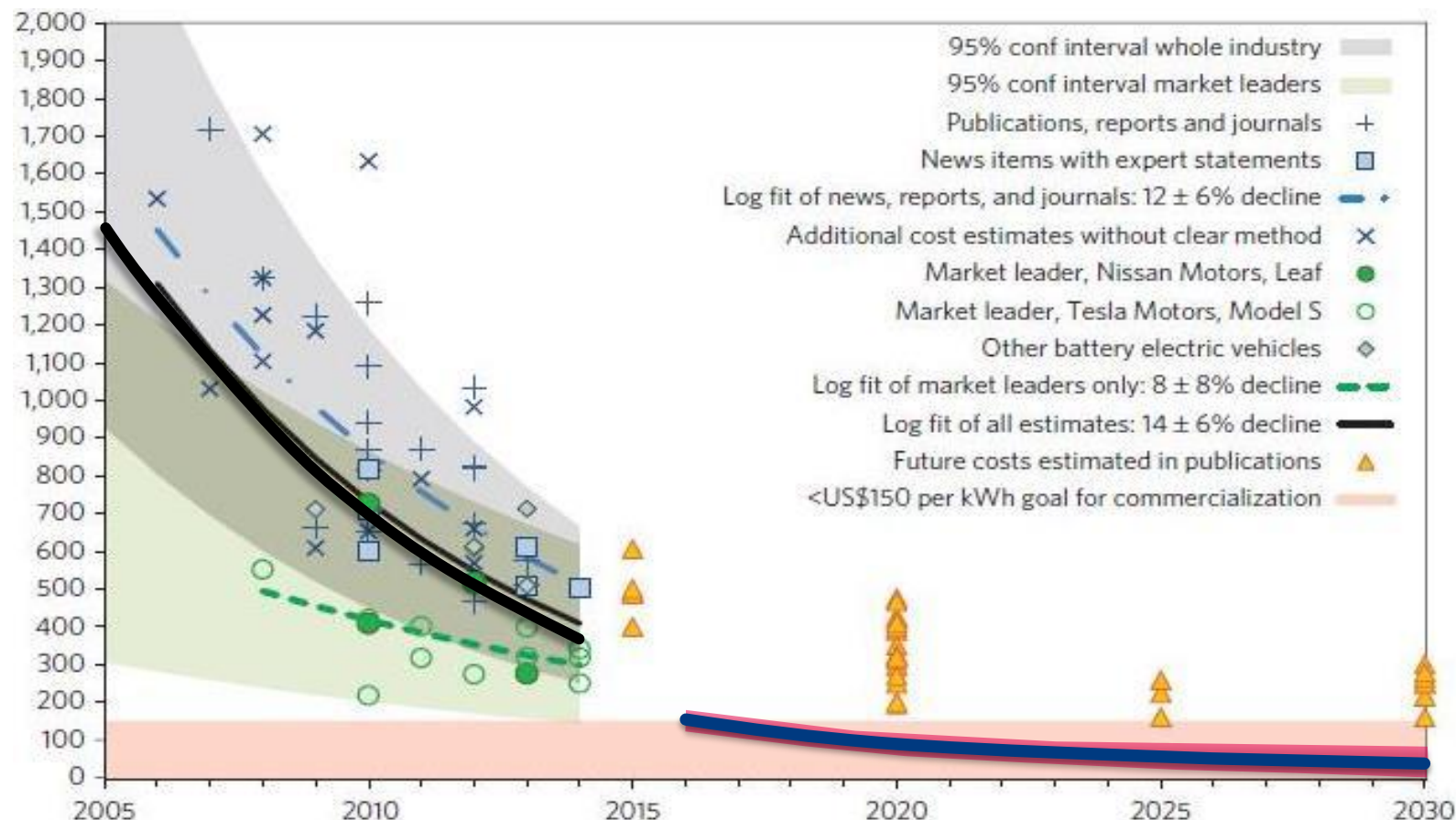


Did you know there is already enough discovered lithium to make 4 *billion* cars? (with 65 kWh batteries)?  
10 kg of recyclable lithium vs 40 tons of gasoline...



**Did you know that over the cars lifetime gasoline and maintenance is already twice as expensive as the battery?**

# Our research indicates storage will become much cheaper still, making the ICE uncompetitive, at least in passenger cars



Source: Björn Nykvist and Mans Nilsson, Nature Climate Change, March 2015 & internship report Anand Lineshsundrani



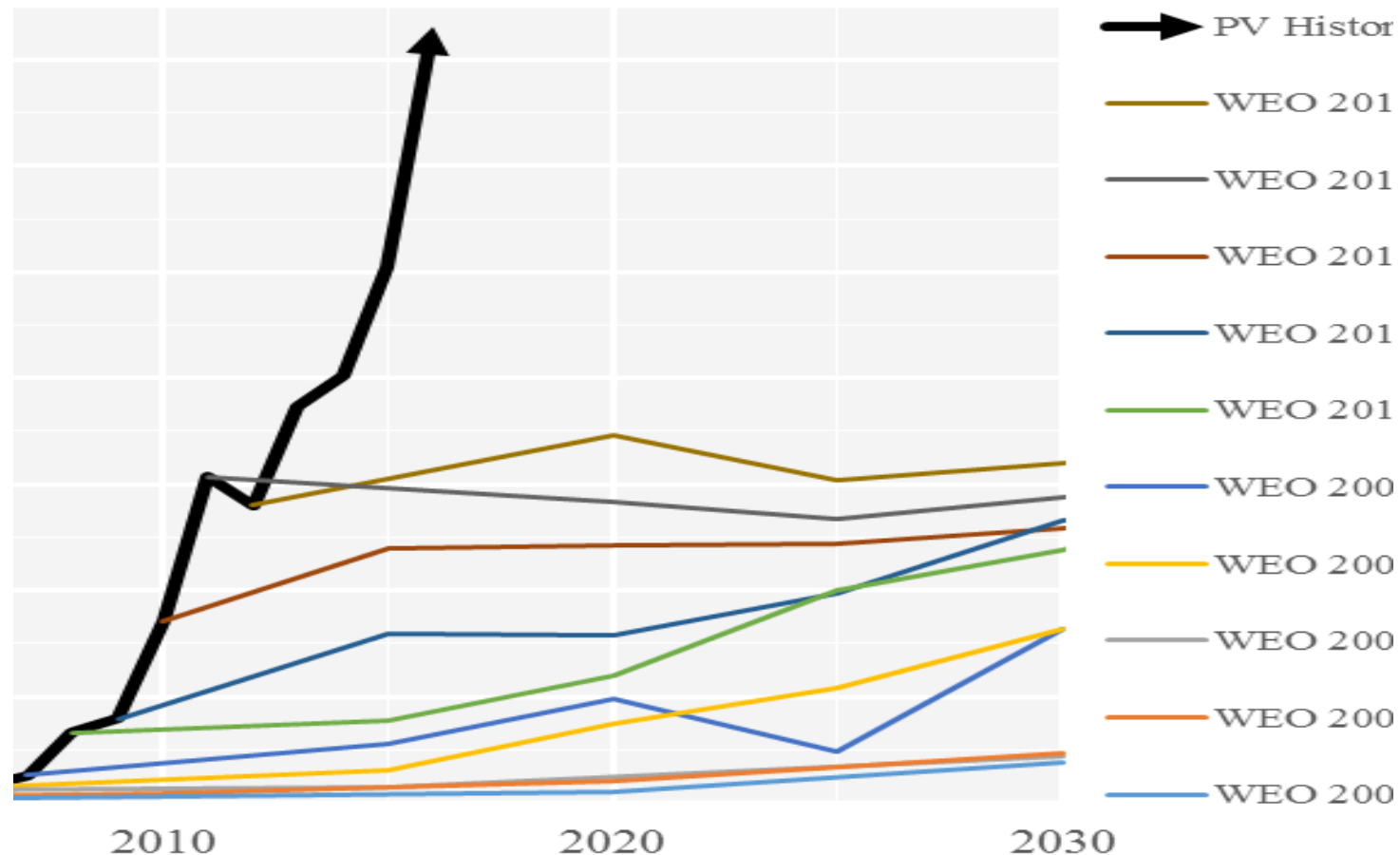
# Wind, solar, EVs, storage...

## This is a perfect storm!



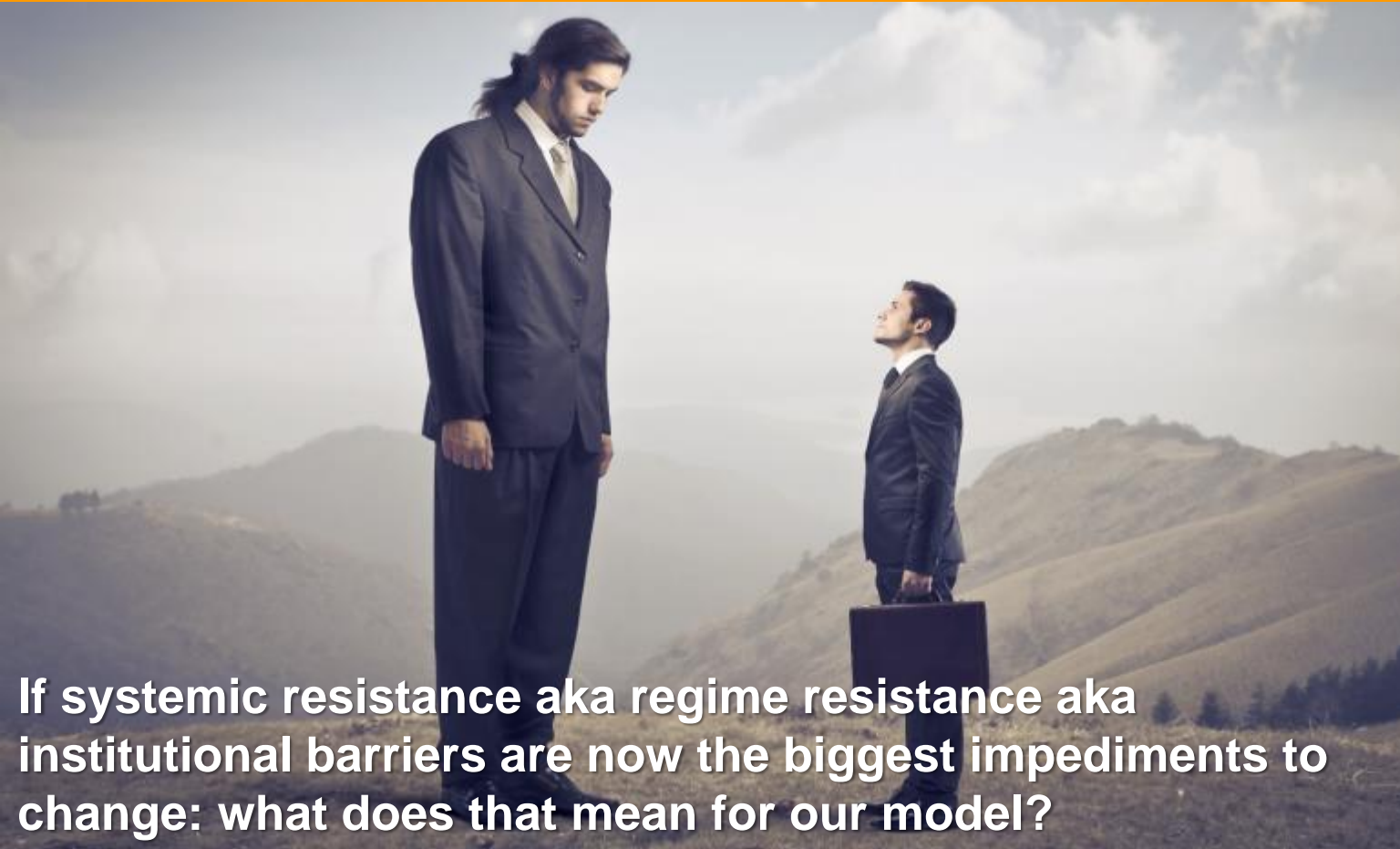
# But what does the worlds most famous model predict? (WEM of IEA)

Files: Historic Data vs IEA WEO Prediction:



# Reminded me of my KPN days

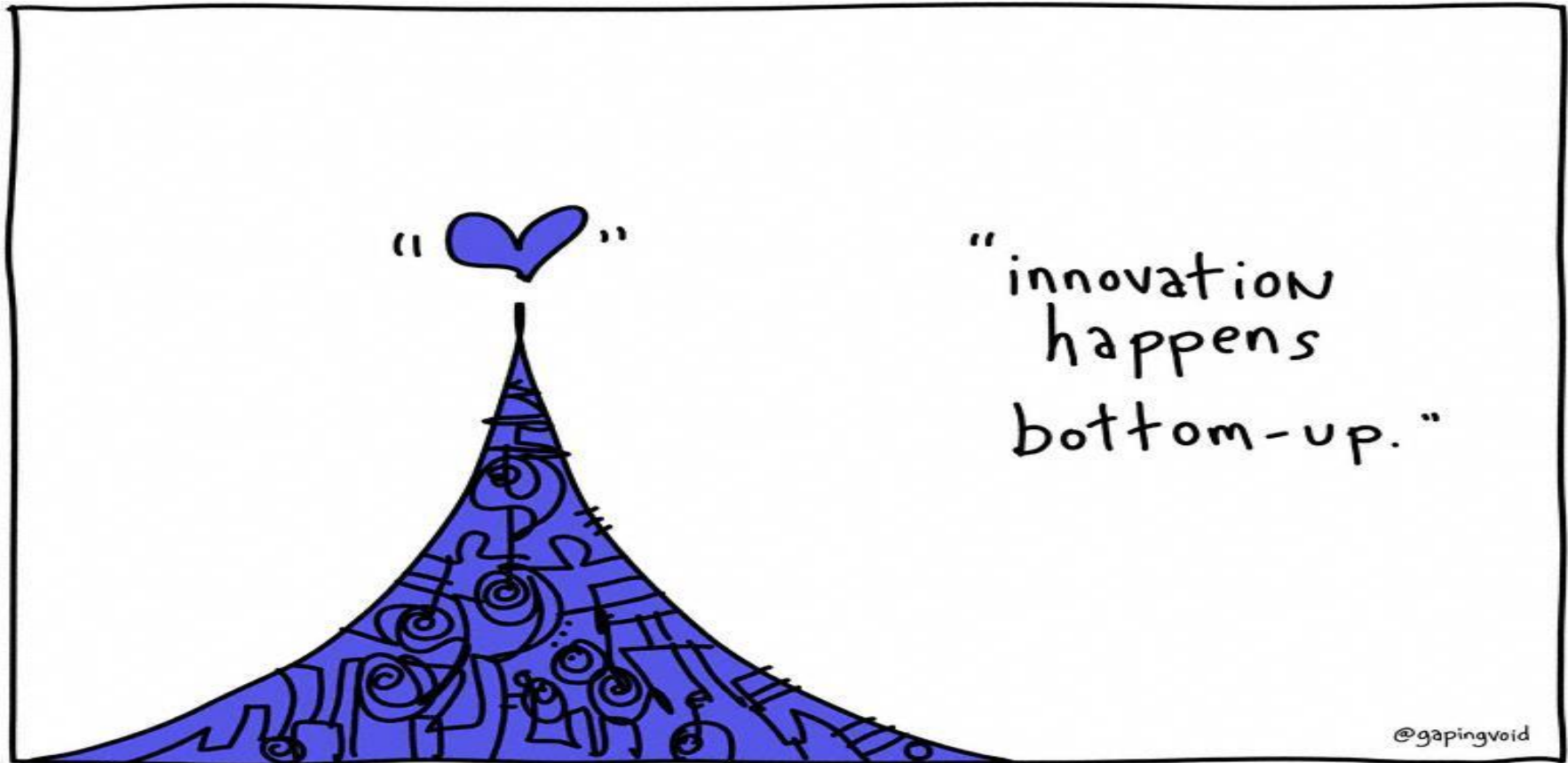
I made money for 25+ years by claiming Internet was going to be big



If systemic resistance aka regime resistance aka institutional barriers are now the biggest impediments to change: what does that mean for our model?



# Big breakthroughs are not directed top-down: we need a bottom-up model



# The questions are interrelated so we need an integral model

- Battery and drive-train prices determine the succes of EVs and the success of EVs determines battery and drive train prices
- The succes of EVs is determined by available models and available models are determined by the succes of EVs
- Driving behavior determines how interesting an EV is, where you need charge points and what room you have for smart charging.
- The availability of charge points co-determines the desirability of EVS which co-determines the need for charge points (chicken-egg problem).

# The developments play on different levels so we need a multi-level model

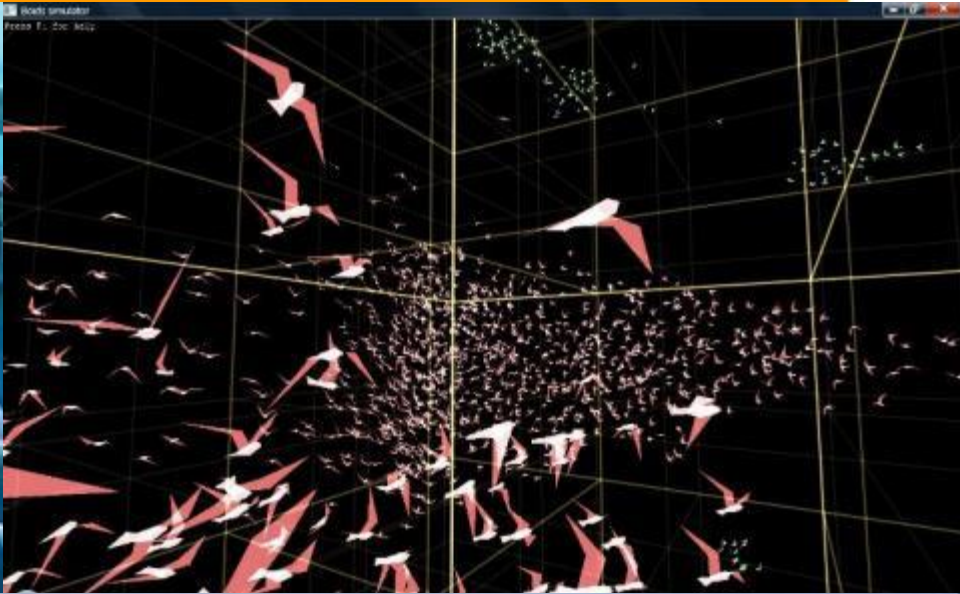
- (Inter)national to model climate problems and technological advances (batteries, drivetrains, renewable energy generation)
- Regional to model driving to and from destination and the required charge points
- Individual to simulate buying/charging decisions and model the load on the grid



We must be able to tell quantified narratives:  
combining numbers with stories is the only  
way to make sense of so much complexity



# Agent-based modeling

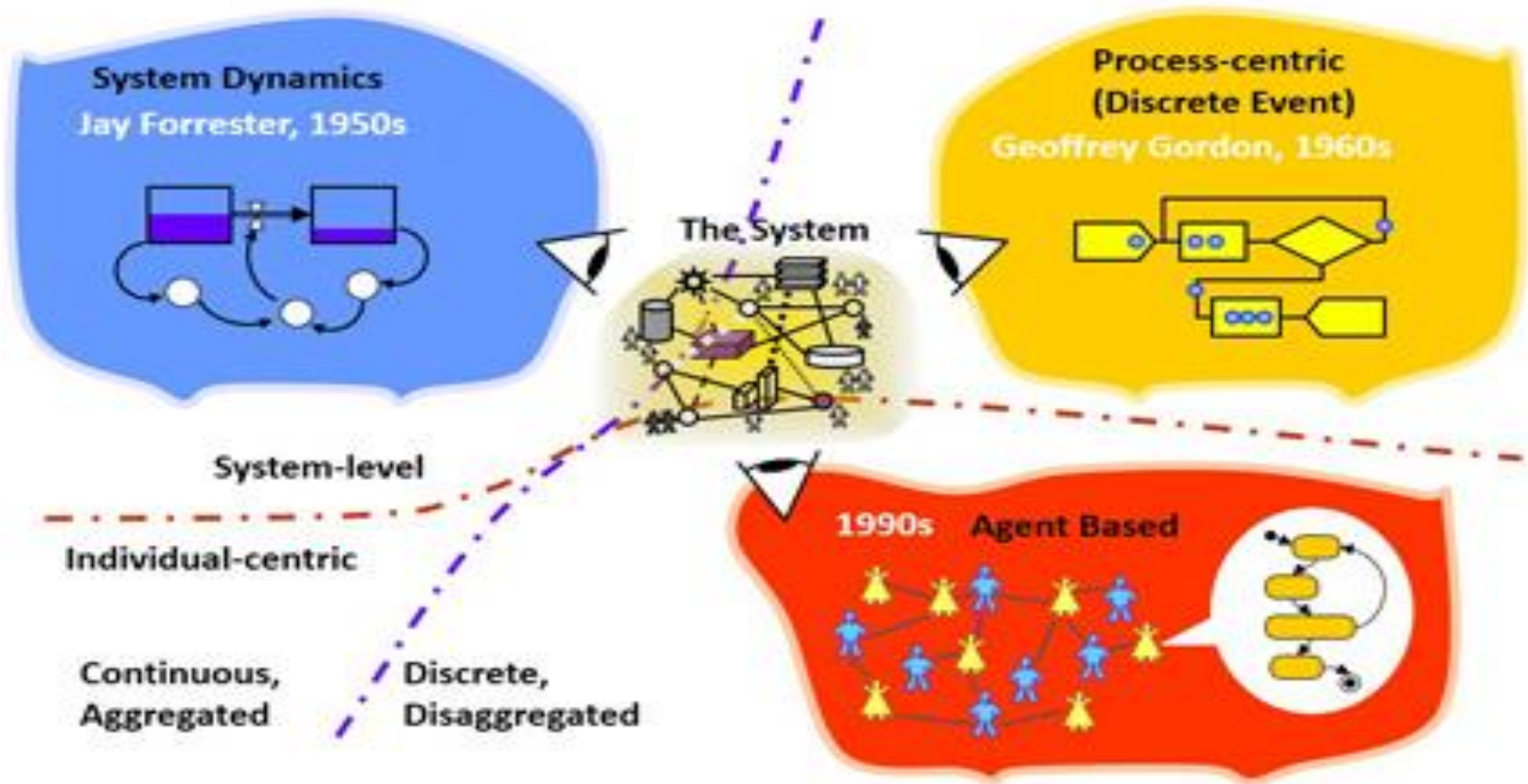


# Why agent-based modeling?

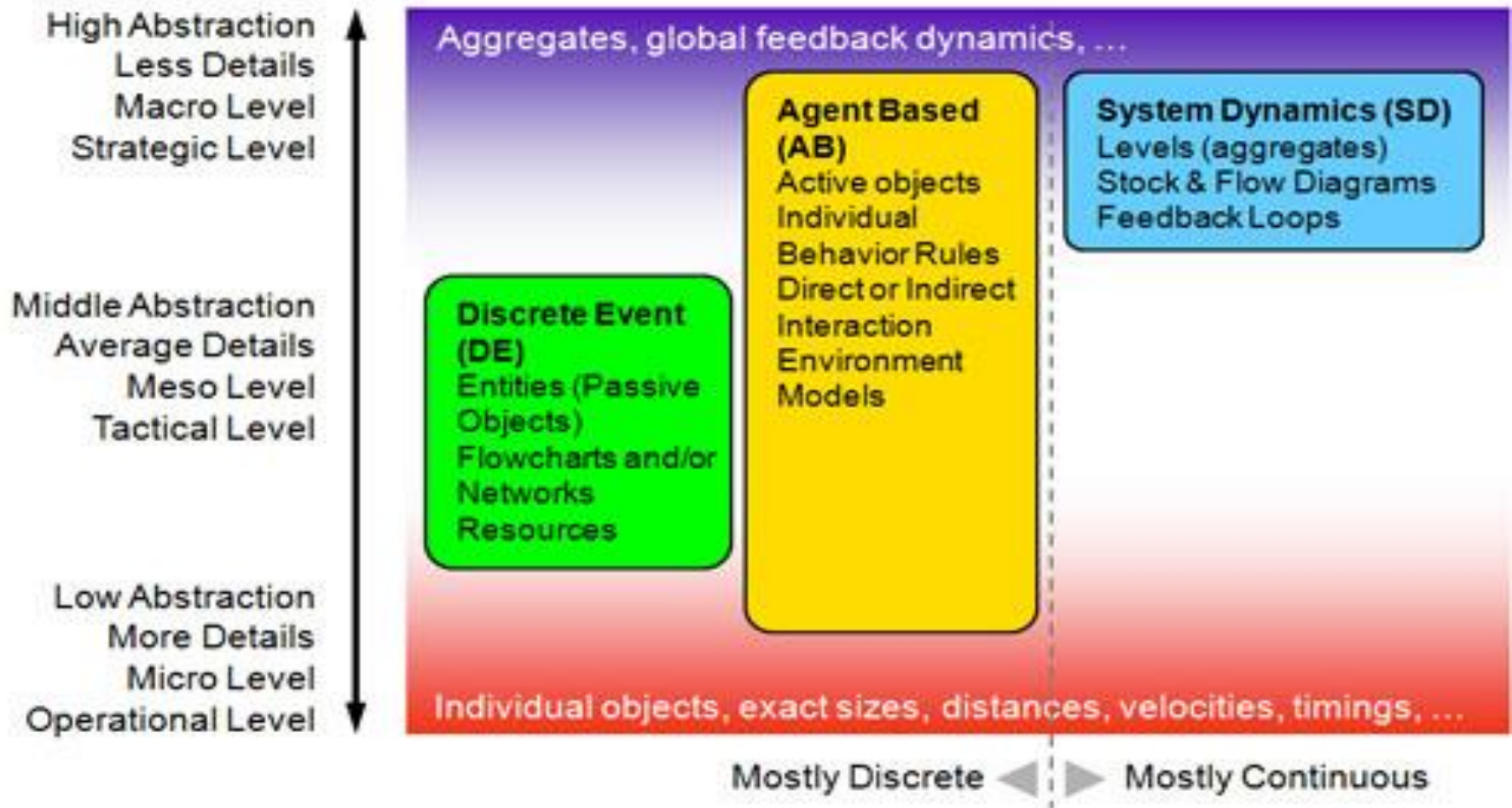
- **Bottom-up without imposing the structure of the system in advance (no pre-determined feedback loops)**
- **Ability to divide the complexity into self contained "agents" makes it manageable (linear instead of exponential growth of complexity when you add variables)**
- **Using recognizable agents and a bottom-up approach means we can enlist the help of domain experts**



# Bottom up and actor based



# From low to high abstraction level



# What ABM tool to use?





# Many Java based frameworks

- Possible but only programmers and productivity modeler is low
- Less useful for domain experts and social scientists
- Python is in-between-solution (but execution speed)

[illegible]

# Anylogic is expensive (or limited) and you end up programming in Java

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a statechart model for a person's health and weight. The model includes states for 'NonDiabetic' (green), 'Diabetic' (red), 'NormalAndUnderweight' (yellow), 'Overweight' (yellow), and 'Obese' (yellow). Transitions between these states are labeled with events such as 'becomingOverweight', 'losingOverweight', 'becomingObese', and 'losingObesity'. The 'Person' object is shown as an active object class with properties like 'color', 'diabetesStatechart', and 'weightStatechart'. The 'becomingOverweight - Transition' properties window is open, showing the 'Name' as 'Person' and the 'Agent' checkbox checked. The 'Person - Active Object Class' properties window is also open, showing the 'Name' as 'Person' and the 'Agent' checkbox checked. The 'Person' object is shown as an active object class with properties like 'color', 'diabetesStatechart', and 'weightStatechart'.

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects

- SimpleInteractingWeightAndDiabetesStatechart
- Main
- Person
- Simulation: Main

Palette

- Statechart
- Statechart Entry Point
- State
- Transition
- Initial State Pointer
- Branch
- History State
- Final State

Properties

becomingOverweight - Transition

Name: Person

Triggered by: Person

Rate: 1

Action:

Person - Active Object Class

General

Name: Person

Ignore

Agent

Startup code:

Destroy code:

# Game engines like unreal are all about how it looks

```
void CObject::OnDestruct()
{
    // TODO: Add your specialized code here and/or call the base class
    Super::OnDestruct();

    if (IsA(UClientState)) {
        UClientState* ClientState = Cast<UClientState>(this);
        ClientState->Release();
    }

    if (IsA(UClientStateHolder)) {
        // We need to emit a close notification
        UClientStateHolder* ClientStateHolder = Cast<UClientStateHolder>(this);
        // Emit the close notification (before a UClientState is released)
        ClientStateHolder->OnClose();
        ClientStateHolder->Release();
    }

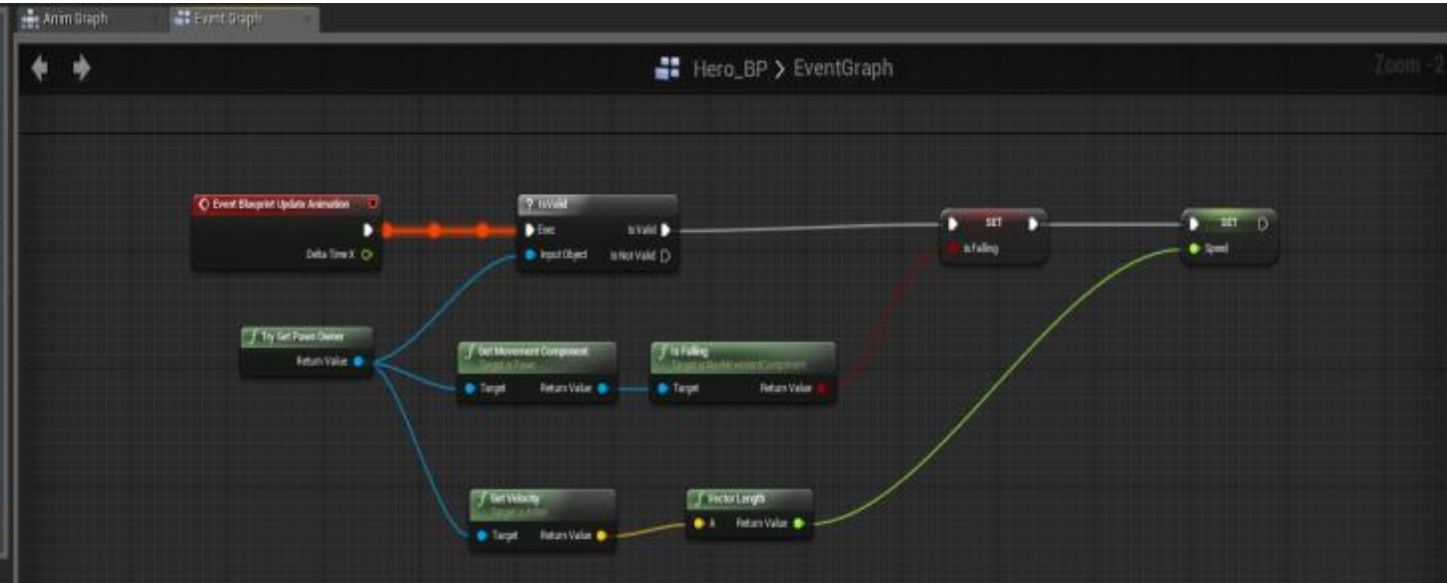
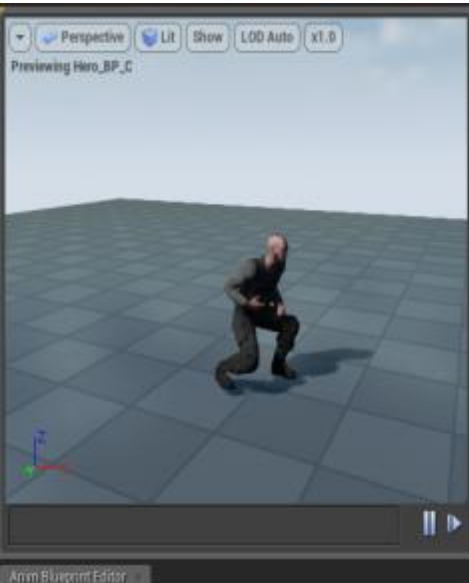
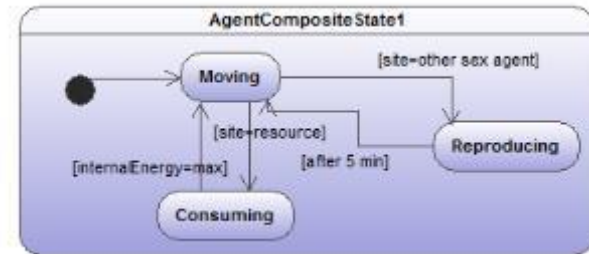
    // Disconnect the object
    UClientState* ClientState = Cast<UClientState>(this);
    ClientState->Release();
    // This is not intended to close client references
    ClientStateHolder->Release();

    if (IsA(UClientStateHolder)) {
        UClientStateHolder* ClientStateHolder = Cast<UClientStateHolder>(this);
        ClientStateHolder->Release();
    }

    if (IsA(UClientStateHolder)) {
        UClientStateHolder* ClientStateHolder = Cast<UClientStateHolder>(this);
        ClientStateHolder->Release();
    }

    if (IsA(UClientStateHolder)) {
        UClientStateHolder* ClientStateHolder = Cast<UClientStateHolder>(this);
        ClientStateHolder->Release();
    }

    // Remove internal state
    InternalState = nullptr;
    if (InternalState) {
        delete InternalState;
    }
}
```





# Netlogo: most popular ABM around and clever use of DSL (like Matlab and R)

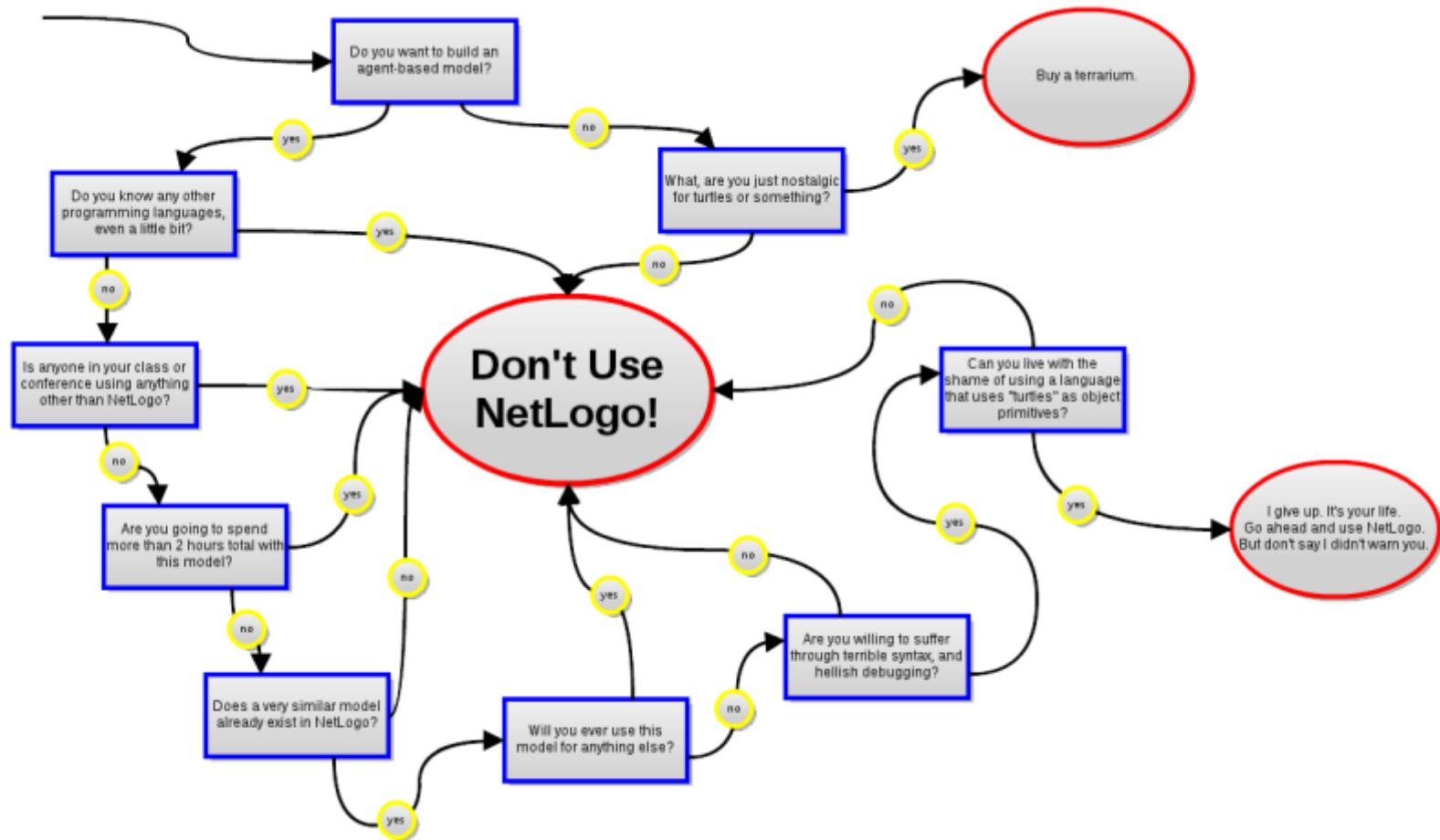
The image shows the NetLogo Wolf Sheep Predation model interface. The left pane displays the code editor with the following code:

```
globals [grass] ;; keep track of how much grass there is
;; Sheep and wolves are both breeds of turtle.
breed [sheep a-sheep] ;; sheep is its own plural, so v
breed [wolves wolf]
turtles-own [energy] ;; both wolves and sheep have energy
patches-own [countdown]

to setup
  clear-all
  ask patches [ set pcolor green ]
  ;; check GRASS? switch.
  ;; if it is true, then grass grows and the sheep eat it
  ;; if it is false, then the sheep don't need to eat
  if grass? [
    ask patches [
```

The right pane shows the model's interface. It includes a 'setup' button, a 'go' button, and a 'show-energy?' switch. Below these are sliders for 'grass-regrowth-time' (set to 30) and 'grass?' (set to 'On'). There are also sliders for 'initial-number-sheep' (100), 'initial-number-wolves' (50), 'sheep-gain-from-food' (4), 'wolf-gain-from-food' (20), 'sheep-reproduce' (4%), and 'wolf-reproduce' (5%). A table shows the current counts: sheep (146), wolves (104), and grass / 4 (248). A 'populations' graph shows the population of sheep (blue line), wolves (red line), and grass / 4 (green line) over time (0 to 126). The main window displays a 3D view of the simulation area with green grass, white sheep, and black wolves. The 'Command Center' at the bottom shows the prompt 'observer>'.

# Attitude of ICT people towards Netlogo



# ABCD model uses GAMA

# GAMA

1.6.1

A spatially explicit multi-scale agent-based simulation platform



©2007-2014 IRD UMMISCO & Partners

<http://gama-platform.googlecode.com>



# GAMA interface

Language comparable to Netlogo but more OO (Java based)

Uses Eclipse IDE (very powerful debugging / collaboration)

The screenshot displays the GAMA interface for the SparkCity model. The main window shows the GAMA code editor with the following code:

```
/* GLOBAL VARS */
file shape_file_buildings <- file("../includes/building.shp") ;
file shape_file_roads <- file("../includes/road.shp") ;
geometry shape <- envelope(shape_file_roads) ;
float geography_increase <- 42.5 ;

int nb_humans <- 100 ;
float g_max_kW <- 10.0 ; // was 3.7
float g_w_start <- 0.5 ; /// was 8.5
float g_w_start_sd <- 0.5 ; /// was 2
float g_w_end <- 17.5 ;
float g_w_end_sd <- 2.0 ;
float g_no_charging_start <- nil as float ;
float g_no_charging_end <- nil as float ;
int g_charge_cap <- 1.75*nb_humans as int ;
float g_not_charge_chance <- 0.25 ;
int g_timer <- 100 ;

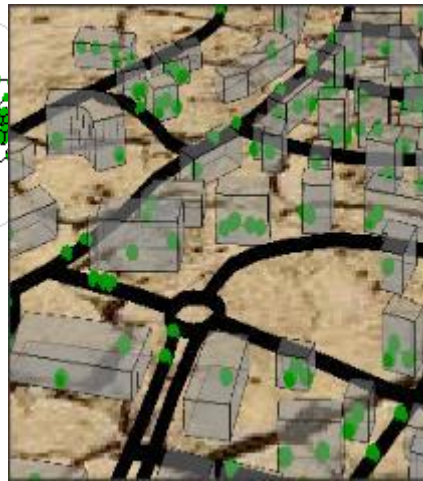
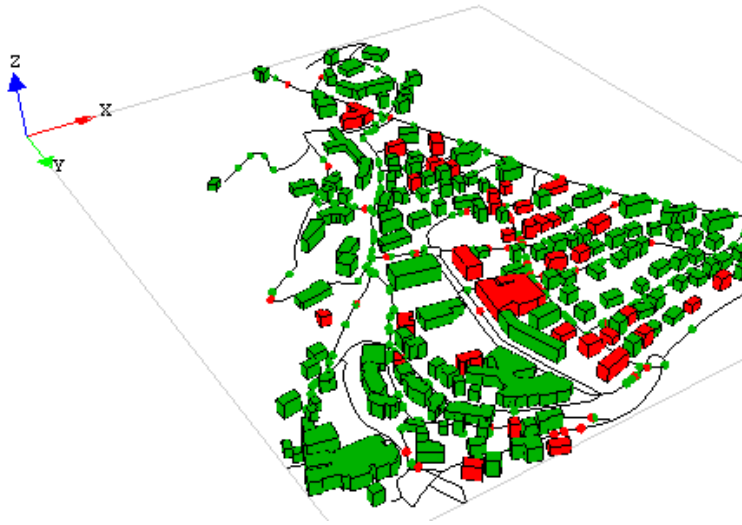
int g_e_use ;
int g_cars_charging ;
int g_cars_connected <- nb_humans ;
list<int> use ;

float step <- 36.0 ;
float hour update: ((time / 36) mod 2400)/100 ;
float g_speed <- 0.3 ;
float g_speed_sd <- 0.15 ;
```

The right panel shows the simulation results. The top chart, titled "Total Power", plots power (0 to 400) against time (0 to 24). It shows a red line for "cars" and a blue line for "total nr of connected cars". The bottom chart, titled "city\_display (Simulation 0 o...", shows a map of the city with a 3D coordinate system (X, Y, Z) and a 2D projection. The status bar at the bottom indicates "31 : 27" and "110M of 856M".

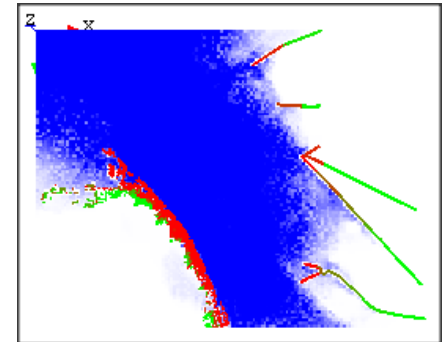
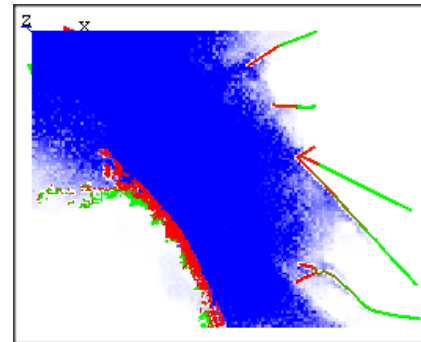
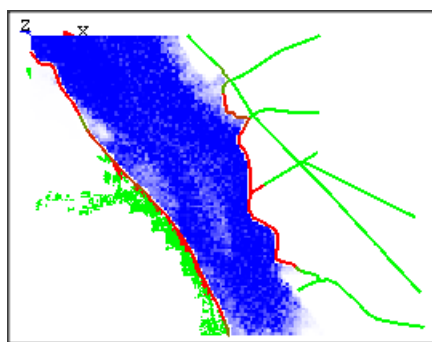
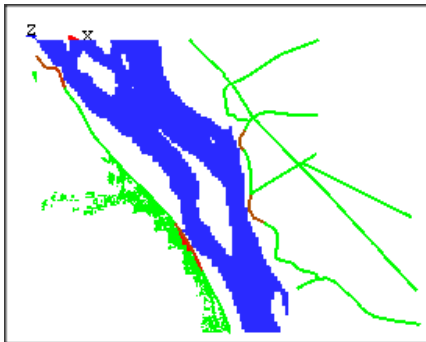
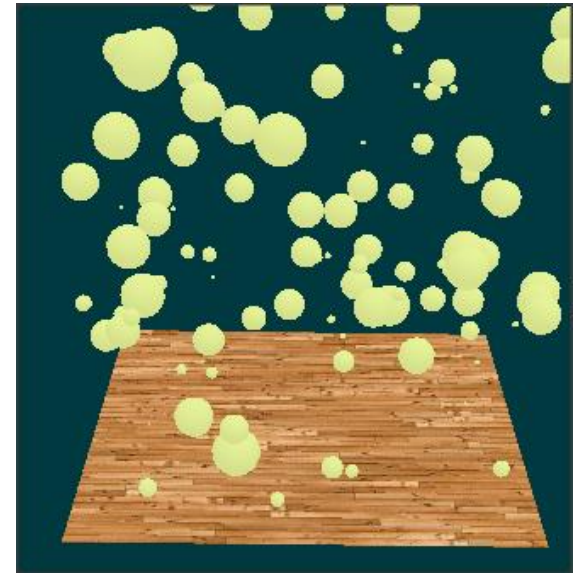
# GAMA and GIS

Agentifying shape files is a breeze



# GAMA facilitates multi-level interaction

- Advanced grouping
- Interaction based on graph, distance, physical properties, etc.
- Detailed driving modules

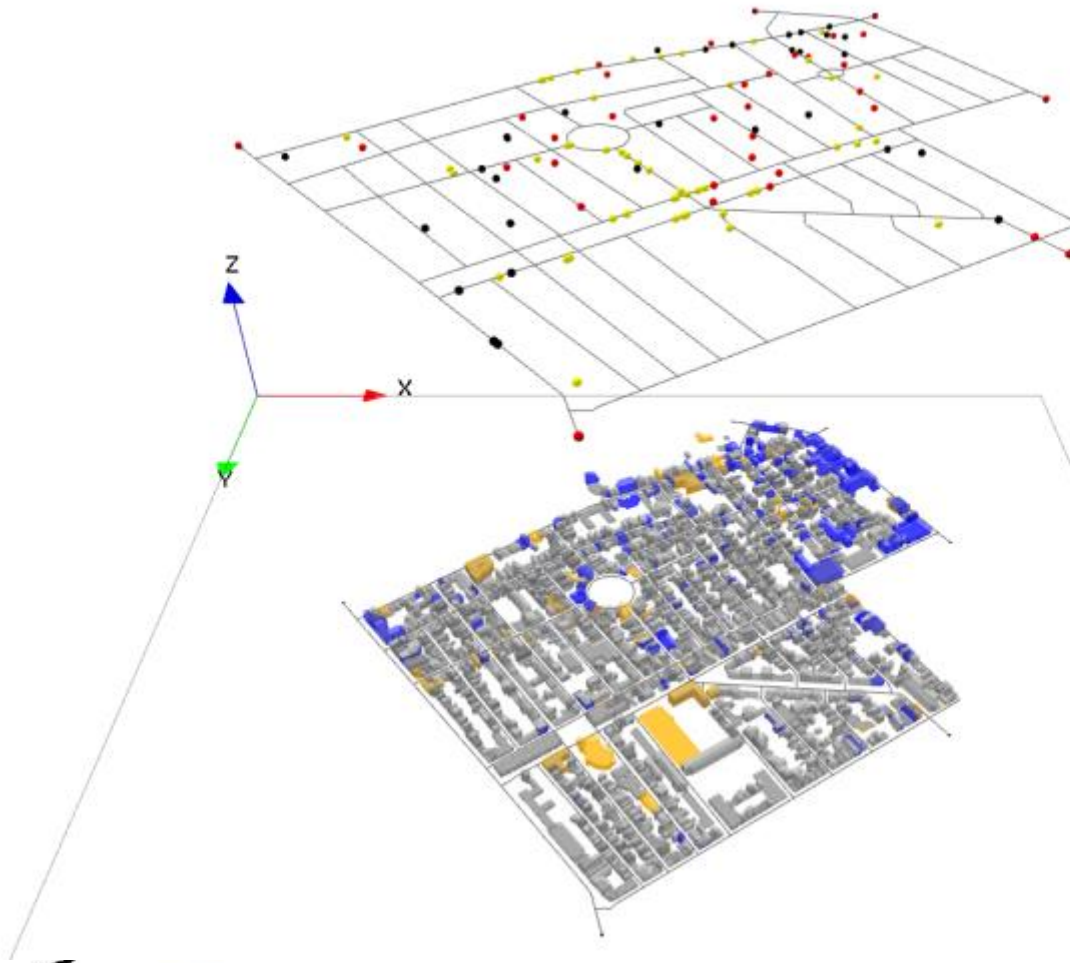




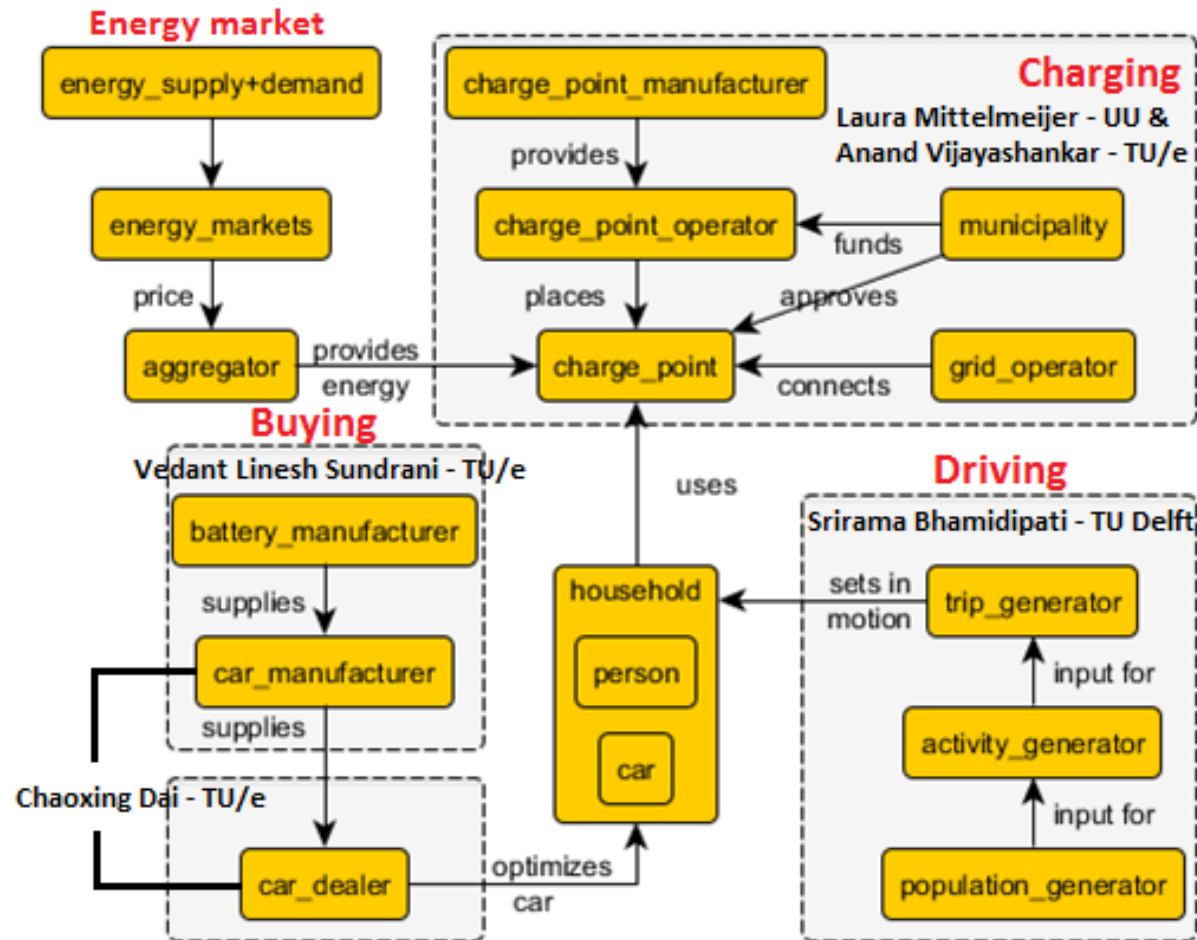
# Where we are now: modeling real neighborhoods



# Add layers with households, EV buying, charging, electricity grid.



# Overall model





# Conclusions

- **Solar, wind, EVs and storage are poised to take over from fossil fuels in a perfect storm**
- **We need actor based, bottom-up, integral, multi-level models to manage and direct this transition**
- **ABM is the perfect modelling paradigm for the energy transition and GAMA is the perfect tool**
- **Our ABCD model is an attempt to create quantified narratives for the energy transition**
- **Starting september we need new students who want to do their master thesis with us (500 euro / month)**