

MESSAGE_{ix} Workshop - Session II

Building an Energy System Model (Part 1)

Energy Program, International Institute for Applied Systems Analysis (IIASA), Austria

MESSAGEix Workshop (online), 8 Sep 2020

The MESSAGE_{ix} workshop team, Sep 2020



Behnam Zakeri



Paul Kishimoto



Oliver Fricko



Francesco Lovat



Muhammad Awais

Energy Program, International Institute for Applied Systems Analysis (IIASA), Austria

The MESSAGEix Modeling Framework

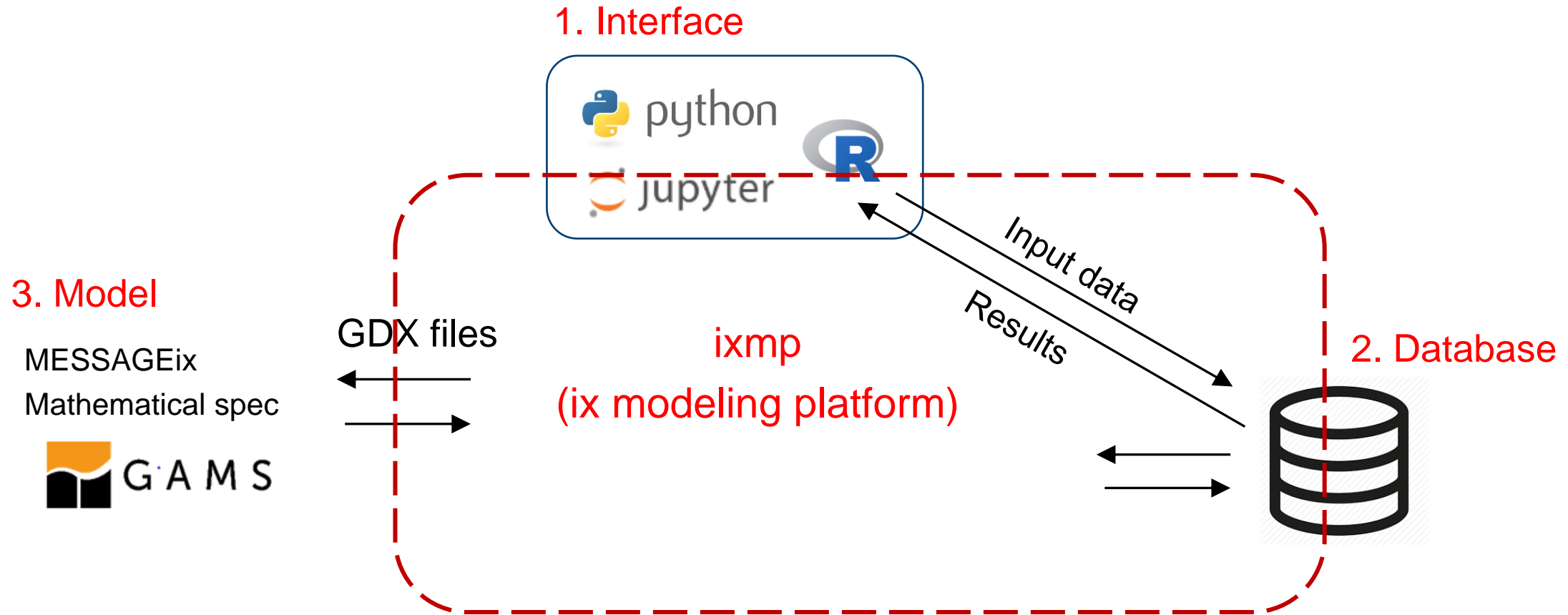
Recap...

- MESSAGEix is an open, version-controlled systems engineering **modeling framework**
- ix modeling **platform** (*ixmp*) is a data warehouse for facilitating high-powered modeling work
- python and R are the main interfaces for modelling using MESSAGEix
- MESSAGEix **mathematical model** is written in GAMS
- Documentation of the MESSAGEix model and tutorials are available online

<https://docs.messageix.org>

The MESSAGE_{ix} framework: Workflow of modeling

Recap...



MESSAGEix framework: Building an energy system – Part 1

Agenda of this Session

- A note on optimization (if needed)
 - MESSAGEix mathematical model and its structure
 - Working with MESSAGEix tutorials: building a simple model
- ➔ Voting feature will be used to measure how much time we should spend

After this tutorial

The goal is to...

- Learn about the logic behind the MESSAGEix energy system model
- Be able to work on a MESSAGEix model using Jupyter Notebook
- Be familiar with basic terminology of a MESSAGEix model

Requirements

- MESSAGEix framework installed and running
- Knowledge on energy systems
- Patience, motivation, and curiosity

Energy Systems

Different scales: community, city, country, region, and global

- A system of energy resources, conversion/processing, transmission and distribution technologies, and services
- Technologies (cars, ...)
- Commodities (oil, gas, ...)
- Sectors (industry, transport, ...)
- Demand services (electricity, lighting, heating, ...)
- Agents (consumers, producers, Operators, etc.)

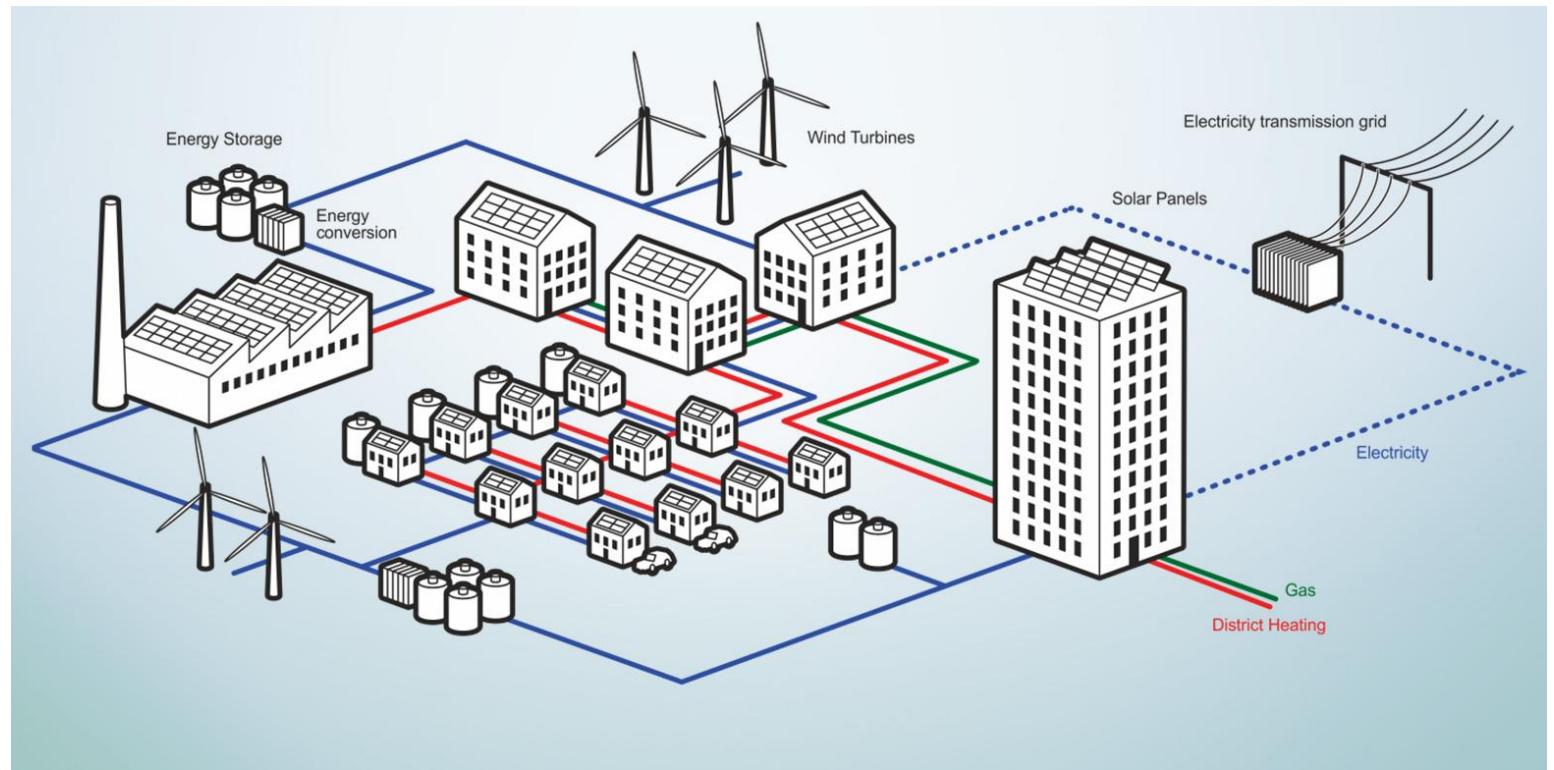


Image: ETH Zurich

Linear programming (LP)

Finding the best (optimal) solution

- The goal is to optimize a linear objective function
- There are a set of *decision variables*
- There are some constraints (bounds on or relationship between decision variables)

$$\text{Maximize} \quad \mathbf{c}^T \mathbf{x}$$

$$\begin{array}{ll} \text{subject to} & \mathbf{Ax} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \end{array}$$

Example: the best way to commute to work

- **Decision variables:** walking, biking, bus, train, taxi, private car, car sharing
- **Objective function:** cheapest or fastest option (least environmental footprint, least walking option)
- **Constraints:** maximum 2 hours commute/day, maximum 300 euro/month, no later than 7 PM, ...
- **Feasible region:** usually there many alternative solutions but not all of them are feasible

Linear programming (LP) (reminder)

Applications of LP

- Production management
- Personnel management
- Marketing management
- Resource/ inventory management
- Blending problem, etc

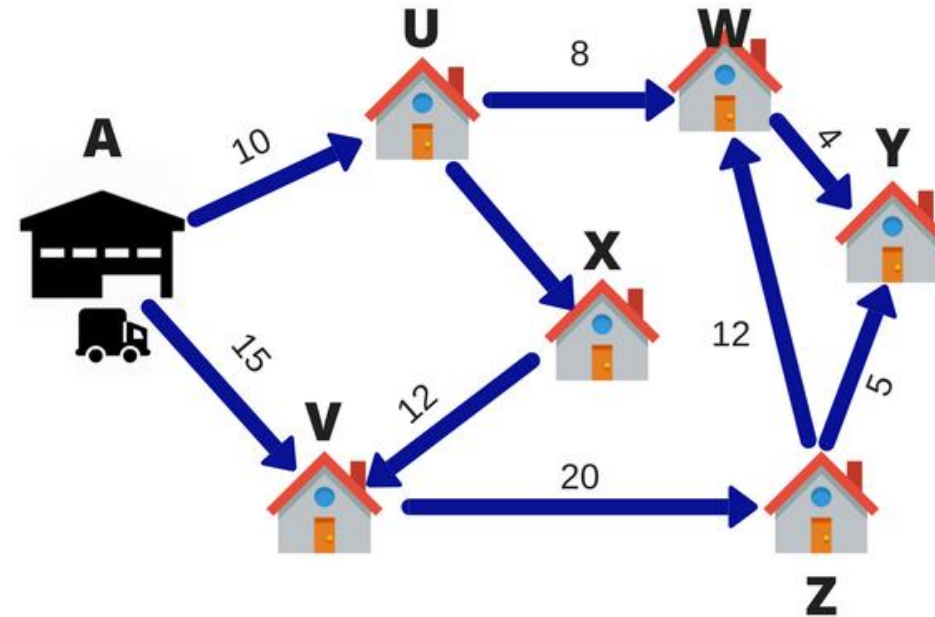


image: deepai.org

Principle:

Maximizing the utility (use of resources)

Or Minimizing the cost of meeting a service → MESSAGEix

MESSAGEix: a model for investment and planning

Minimizing total discounted cost of the system

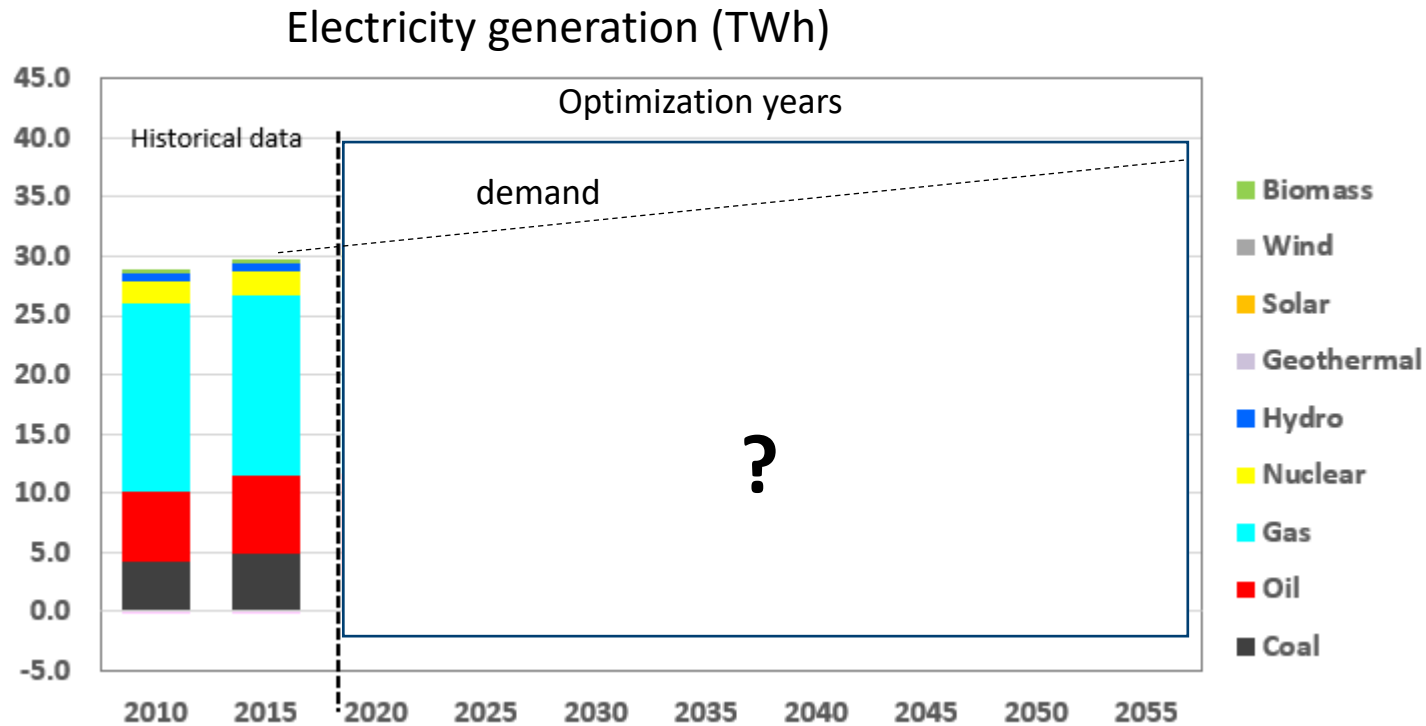
- **Objective:** The least cost option for meeting certain services (demand) $\rightarrow \min cT \cdot x$
- **System:** a network of technologies (processes), resources, and commodities (products)
- **Cost of the system:** installing/maintaining **capacity**, cost of **activity** (O&M), taxes, emission penalties, land use costs (if any), etc.
- **Constraints:** maximum use of a technology, growth/decline rates of activity, capacity factor, etc.
 $\rightarrow s.t. A \cdot x \leq b$

A note on “capacity” and “activity” (MESSAGEix formulation)

- **Capacity:** installed units of a technology (e.g. 150 MW power plant)
- **Activity:** operation of that technology (e.g., 800 GWh)
- **Reminder:** capacity \neq activity capacity * capacity factor = activity

The MESSAGE_{ix} framework : Investment planning

From historical activity/capacity to model years

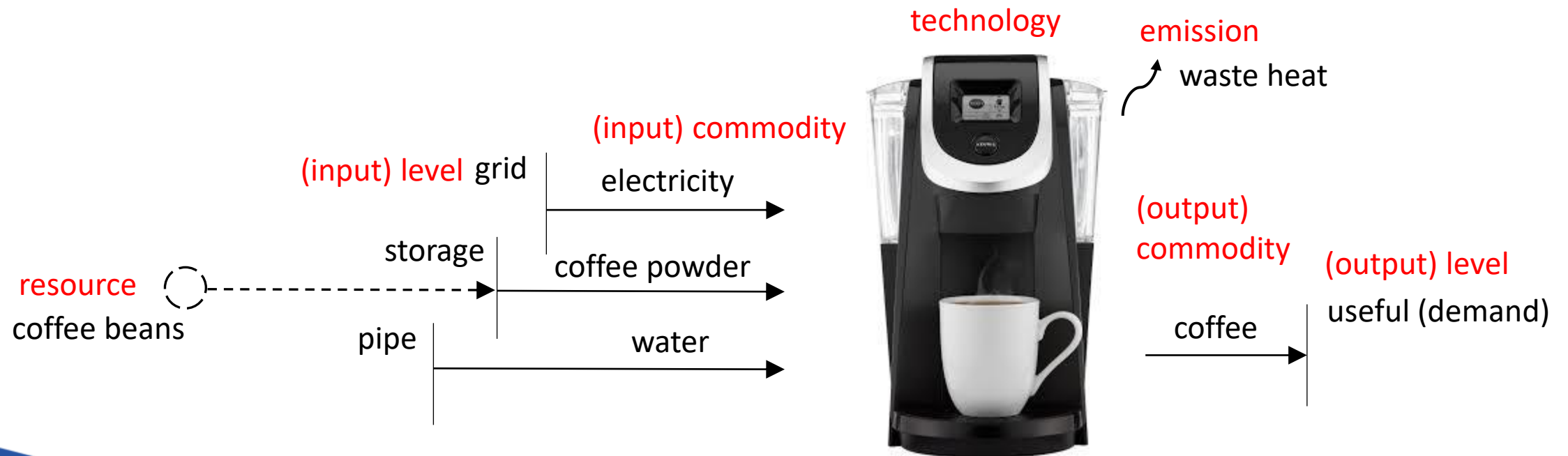


MESSAGEix: a technology-based model

Technologies and resources meet demand

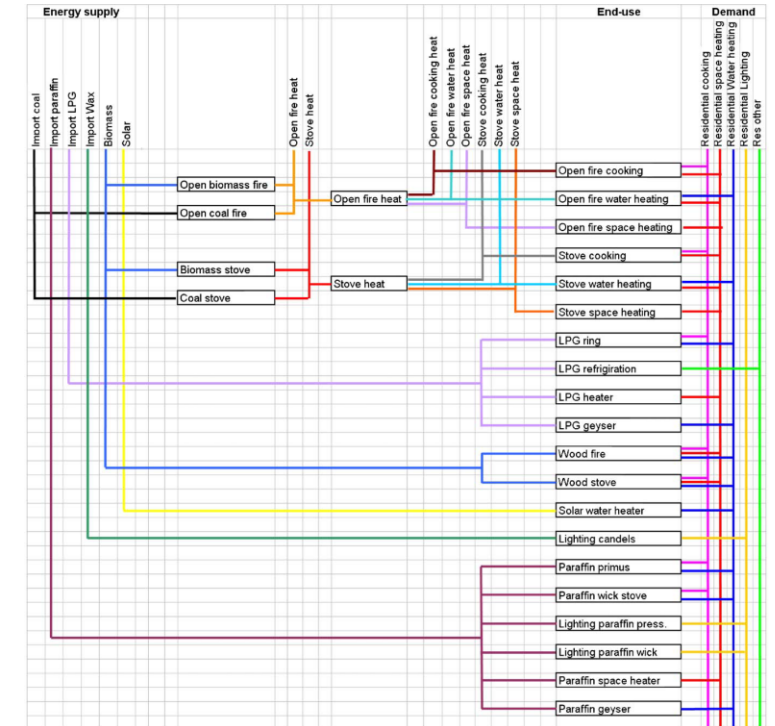
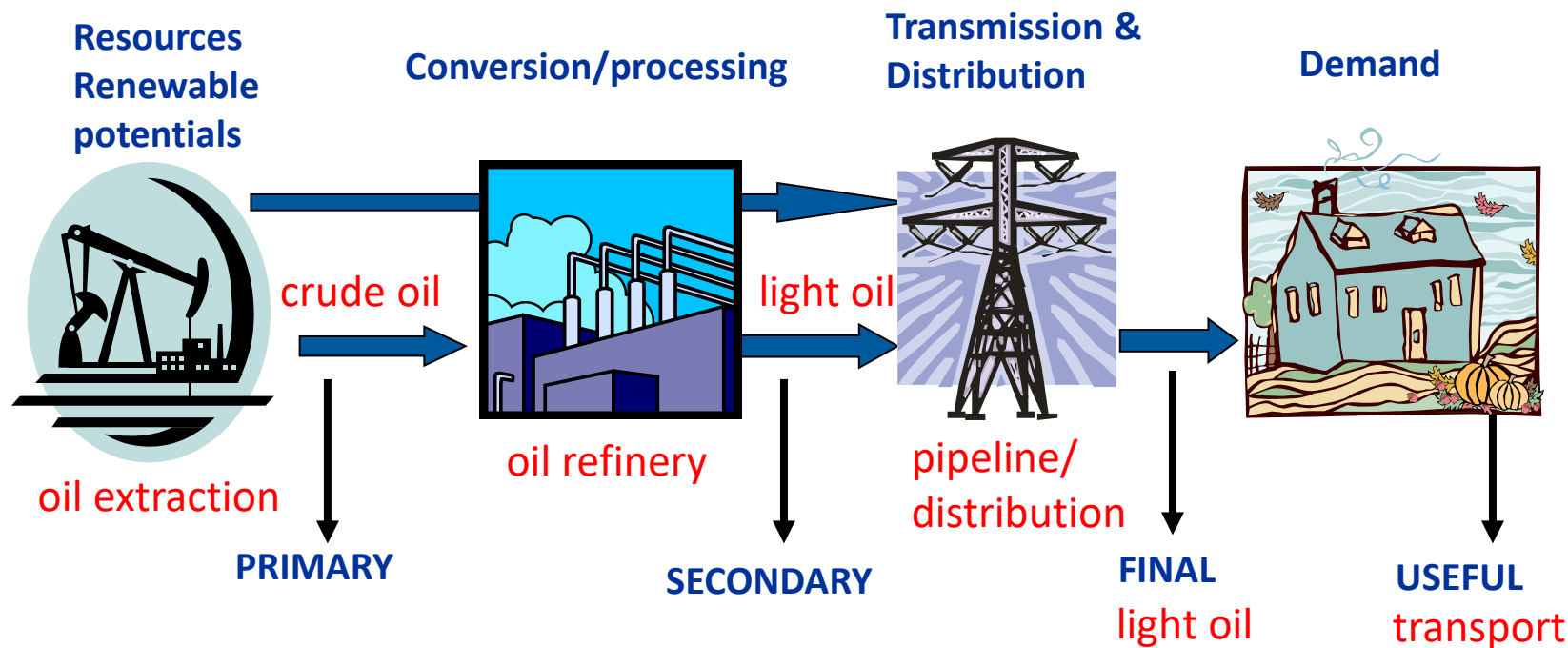
- Example **technologies**: electric car, reactor, pipeline, power plant, building, ship, industrial process

A sample technology: coffee machine



MESSAGEix: a technology-based model (2)

A flexible representation of processes technologies and interlinkages



MESSAGEix: Mathematical notation (GAMS)

Sets:

- List of elements for building a model
- Example: technology, commodity, node, emission

set

members

technology

Solar PV, pipeline

commodity

gas, heat, steel,

Parameters:

- Defining quantities (specification), *e.g., lifetime, efficiency, costs*
- Defining relationships between sets, *e.g., input and output of technology*
- Defining constraints, *e.g., bounds and growth rates*

EQUATIONS:

- Relationship between sets, parameters, **VARIABLES**, etc.
- Building the model

VARIABLES:

- decision variables to meet the objective and constraints, *e.g., ACT, CAP*

Module Data Sheet M60

ELECTRICAL SPECIFICATION (STC)			
Model	SEAMON-200	SEAMON-250	SEAMON-270
Rated Power (P _{mp})	200W	250W	270W
Rated Current (I _{mp})	8.24A	8.80A	8.75A
Rated Voltage (V _{mp})	24.3V	28.5V	31.1V
Short Circuit Current (I _{sc})	8.86A	9.07A	9.05A
Open Circuit Voltage	37.8V	38.2V	38.5V

TEMPERATURE COEFFICIENT			
Nominal Operating Cell Temperature	NOCT	°C	46±2
Temperature Coefficient of I _{sc}	α	%/°C	+0.06
Temperature Coefficient of V _{oc}	β	%/°C	-0.38
Temperature Coefficient of P _{mp}	γ	%/°C	-0.49

Technical Data for all Modules

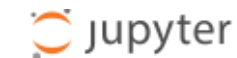
MECHANICAL SPECIFICATION	
Cell Type	Polycrystalline / Monocrystalline / Bifacial
Cell Dimension	156mm x 156mm (6" x 6")
Module Dimension	1650mm x 990mm x 35mm (65" x 39" x 1.4")
Weight	20kg
Front Glass	2mm tempered AR glass
Back Glass	2mm tempered glass

LIMITS	
Operational Temperature	°C -40~+85
Maximum Static Load	Pa 5400
Maximum Wind Load	Pa 2400
Maximum System Voltage	VDC 1500 UL-680
Maximum Series Fuse Rating	A 15



- Locate your tutorial folder in your machine
- Then, open a command window and call *jupyter notebook*
- Navigate to the folder for Westeros tutorials and open the baseline

This tutorial is based on the country of Westeros from the TV show "Game of Thrones".



Select items to perform actions on them.

16 October 2020

Building a MESSAGEix model

Different steps of modeling

- Creating a new scenario (or loading an existing one)
- Declaring required sets (*node, technology, commodity, level, etc.*)
- Defining required parameters (adding numeric data, relating sets to each other, etc.)
 - *demand*
 - *techno-economic parameters (lifetime, efficiency, investment cost, O&M cost, etc.)*
 - *bounds and dynamic constraints (growth rates, diffusion rates of technologies)*
- Solving the model
- Postprocessing and plotting

Building a MESSAGEix model (2)

Minimum information for building a model

- Sets: *technology, node, commodity, level, mode* (of operation), *year, time* (sub-annual timesteps)
 - ➔ Some sets have default values in the model (for example, sub-annual timesteps *time*: 'year')
- Parameters
 - *demand*
 - *output* (of technologies)
- ➔ In MESSAGEix efficiency of technologies is defined with two parameters: *input, output*

$$\text{eff} = \text{output} / \text{input}$$

Working with MESSAGEix scenarios

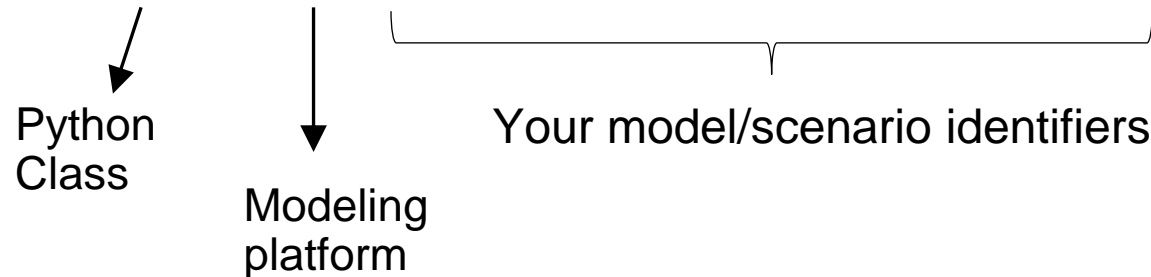
A short note on model/scenarios

- Loading the ixmp platform (connection to the database):

```
mp = ixmp.Platform()
```

- Creating a new scenario:

```
my_scen = message_ix.Scenario(mp, model, scenario, version='new')
```



Example: `model = 'building energy system', scenario = 'baseline' (or 'low efficiency')`

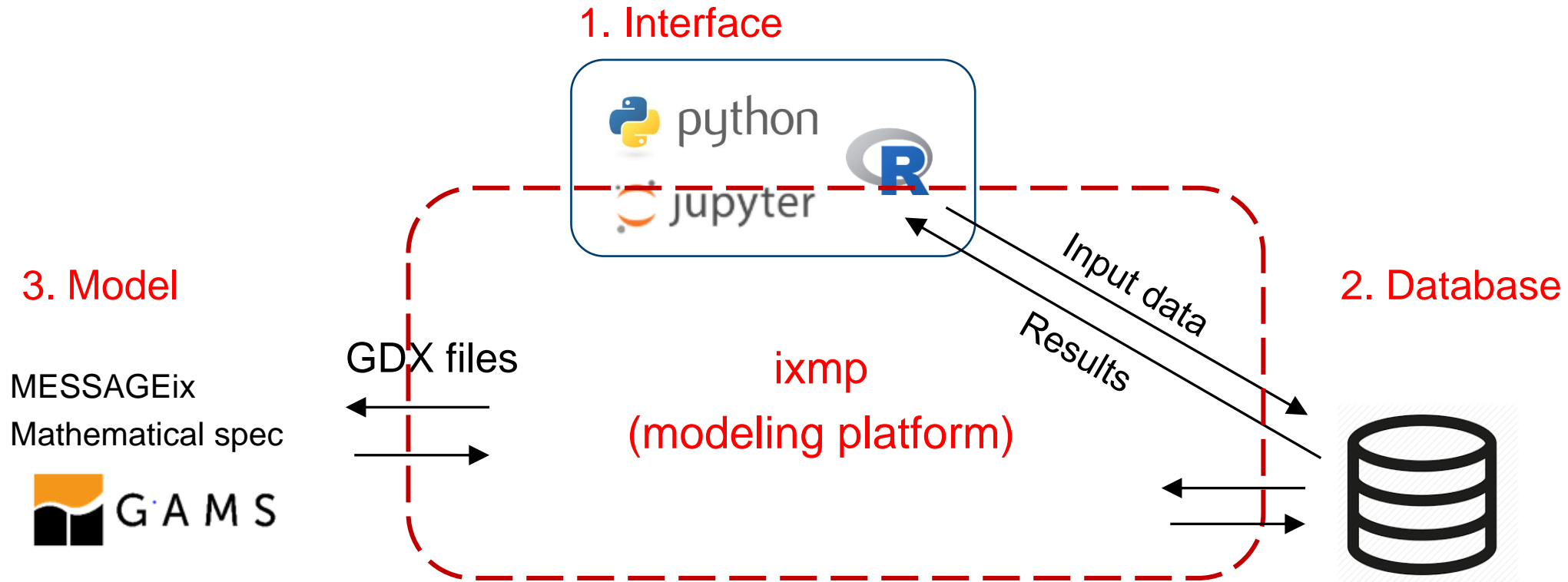
- Loading an existing scenario:

```
my_scen = message_ix.Scenario(mp, model, scenario, version=1)
```

```
my_scen = message_ix.Scenario(mp, model, scenario)
```

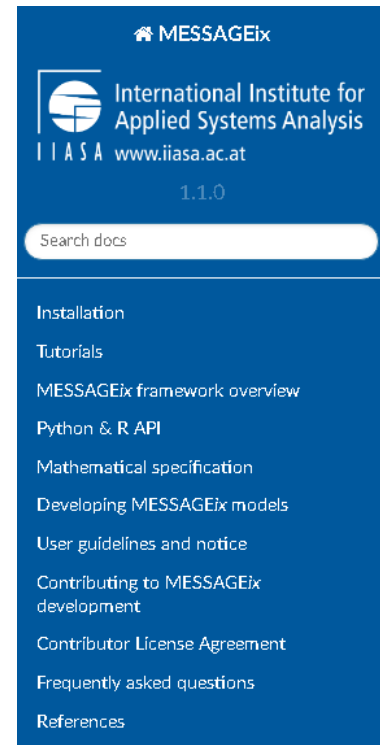
The MESSAGE_{ix} framework : Workflow of modeling

Recap...



MESSAGEix Website: main source of information

- Main page:
⇒ <https://MESSAGE.iiasa.ac.at>
- Open-source GitHub repository:
⇒ https://github.com/iiasa/message_ix
(contribution guide)
- Tutorials and examples



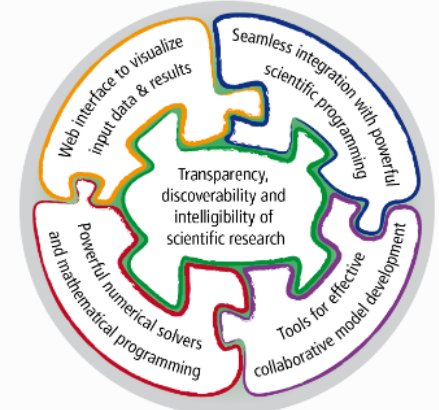
[Docs](#) » The MESSAGEix framework

[View page source](#)

The MESSAGEix framework

Overview and scope

MESSAGEix is a versatile, open-source, dynamic systems-optimization modelling framework. It was developed for strategic energy planning and integrated assessment of energy-engineering-economy-environment systems (E4). The framework can be applied to analyse scenarios of the energy system transformation under technical-engineering constraints and political-societal considerations. The optimization model can be linked to the general-economy MACRO model to incorporate feedback between prices and demand levels for energy and commodities. The equations are implemented in the mathematical programming system **GAMS** for numerical solution of a model instance.



The ix modeling platform (source: [1])

Thank you very much for your attention!

Dr. Behnam Zakeri
Research Scholar – Energy Program
International Institute for Applied Systems Analysis (IIASA)
Laxenburg, Austria
zakeri@iiasa.ac.at

This presentation is licensed under
a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

