

energyRt

energy systems modeling R-toolbox

Oleg Lugovoy
Vladimir Potashnikov

Open Energy Modelling Workshop
October 27-28, Milano 2016

Outline

- Demand for **reproducible research**
- The concept of **object oriented programming (OOP)**
- **energyRt**
 - language
 - main objects
 - main methods
 - Main features

Motivation

- minimize **time** of development and application of RES/BottomUp models,
- boost **learning curve** in energy modeling,
- improve **transparency** and understanding of energy models,
- use power of **open-source** to improve energy models and their application,
- making **reproducible research** accessible in RES-modeling,
- **integration** with other models and software.

General modeling workflow

- Model development
 - Design model structure
 - Data collection, processing
 - Model implementation (software)
 - Calibration of the model
- Model application
 - Design scenarios
 - Adjust, update model
 - Running the model
 - Processing results
 - Analysis of the results
 - Conclusions

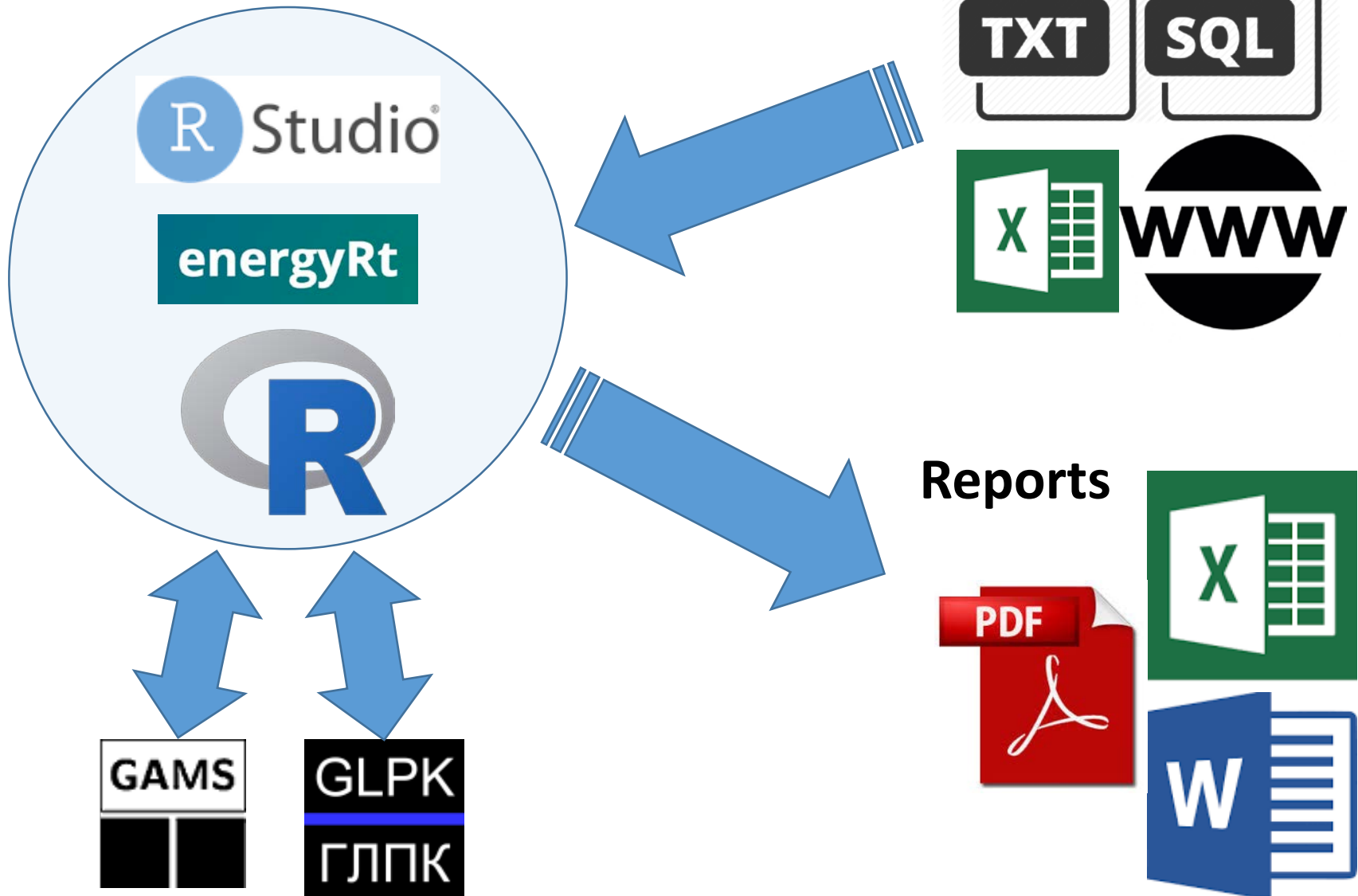
General modeling workflow

- Model development
 - Design model structure
 - Data collection, processing
 - Model implementation (software)
 - Calibration of the model
- Model application
 - Design scenarios
 - Adjust, update model
 - Running the model
 - Processing results
 - Analysis of the results
 - Conclusions

Demand for reproducible research in energy modeling: an example

- How many model runs have your team done to reach “reasonable” results?
- How much time have you and your team spent for processing your model results and finding inconsistencies?
- How much time will it take to reproduce your own results?
- Will be able other people (new team members) reproduce your research?

energyRt: workflow



General modeling workflow with **energyRt**

- Model development
 - Design model structure
 - Data collection, processing (R, energyRt classes: *commodity*, *technology*, *repository*)
 - Model implementation (energyRt classes: *model*; code: GAMS or GLPK)
 - Calibration of the model (energyRt methods: *draw*, *levcost*, *compare*, ...)
- Model application
 - Design scenarios (energyRt classes: *constrain*, *scenario*)
 - Adjust, update model (R, energyRt)
 - Running the model (energyRt methods: *solve*)
 - Processing results (energyRt methods: *report*, *getData*, ...)
 - Analysis of the results (R graphs; energyRt methods: *plot*, *report*, ...)
 - Conclusions

R + RStudio + energyRt

The screenshot displays the RStudio interface with three main panels: the editor, the environment/history, and the viewer.

Editor Panel: Shows the R script `energyRt_basics.Rmd`. The code defines a model structure for a coal-fired power plant, including units, capacity, activity, and various parameters like `cap2act`, `ceff`, `aeff`, `afa`, `fixom`, and `varom`.

```
179 # unit = "kt"),
180 units = list(capacity = "GW",
181             activity = "PJ",
182             varom = "MRMB/PJ",
183             fixom = "MRMB/GW",
184             invcost = "MRMB/GW"),
185 cap2act = 31.536,
186 ceff = list(comm = c("ELC", "CCSC02"),
187            use2cact = c(0.45, 80)),
188 # aeff = list(acomm = "CCSC02",
189 #            comm = "COA",
190 #            cinp2aout = 0.8 * COA@emis$mean[1]),
191 afa = list(year = c(2010, 2015, 2030),
192          afa.up = c(0.7, 0.65, 0.75)),
193 fixom = list(fixom = 50), # MRMB/GW
194 # varom = list(year = c(2010, 2020, 2030, NA),
195 #             varom = c(1, 1.5, 1.8, NA),
196 #             comm = c(NA, NA, NA, "CCSC02"),
197 #             cvarom = c(NA, NA, NA, convert("PMB/t"))
```

Environment/History Panel: Shows the Global Environment with a list of values:

Values	
CCSC02	Formal class commodity
CH4	Formal class commodity
CO2	Formal class commodity
CO2STORAGE	Formal class technology
CO2tax	Formal class constrain

Viewer Panel: Displays a diagram titled "ECOA Coal-fired power plant". The diagram shows a blue rectangular box representing the plant. A red arrow labeled "COA (PJ)" points into the box from the left. Inside the box, the text "cinp2use = 1" is visible. A red arrow labeled "ELC (PJ)" points out of the box to the right. Above the box, the text "ECOA Coal-fired power plant" is displayed. To the right of the box, the text "use2cact = 0.33" and "cact2cout = 1" is shown.

Console Panel: Shows the output of the R script, including the results of the `draw(ECOA)` function call.

```
2 <NA> 2015 110
3 <NA> 2020 90
4 <NA> 2030 0

Slot "early.retirement":
[1] TRUE

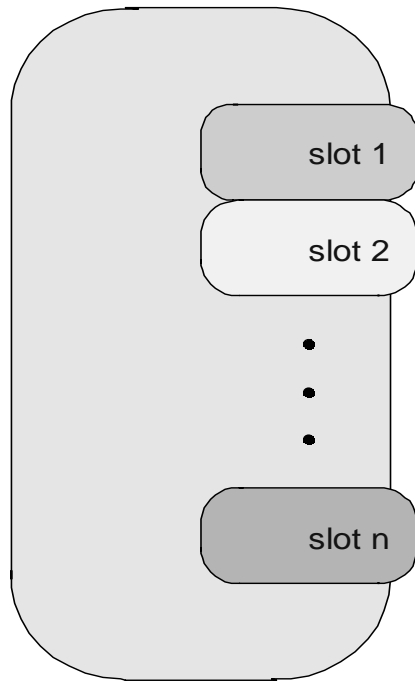
Slot "upgrade.technology":
character(0)

Slot ".S3Class":
[1] "technology"

> draw(ECOA)
```

Classes: custom data structure

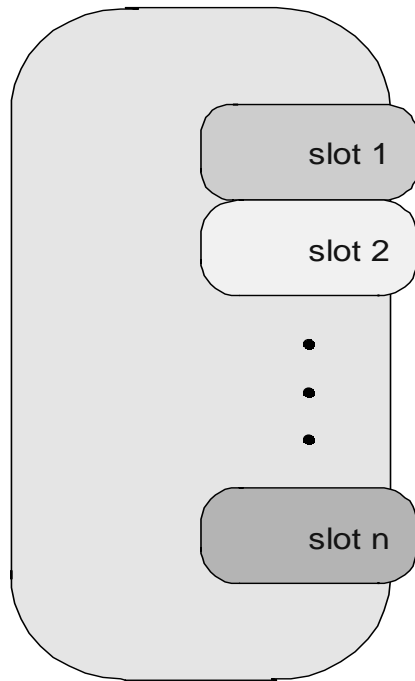
An object/class



Classes: custom data structure (1/2)

Design your own data class

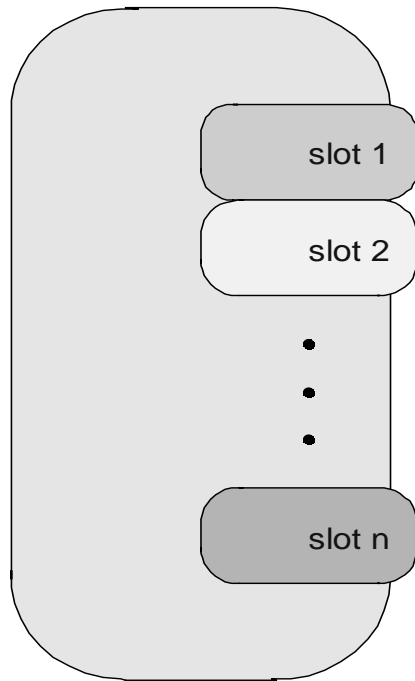
An object/class



Classes: custom data structure (2/2)

Nestable

An object/class



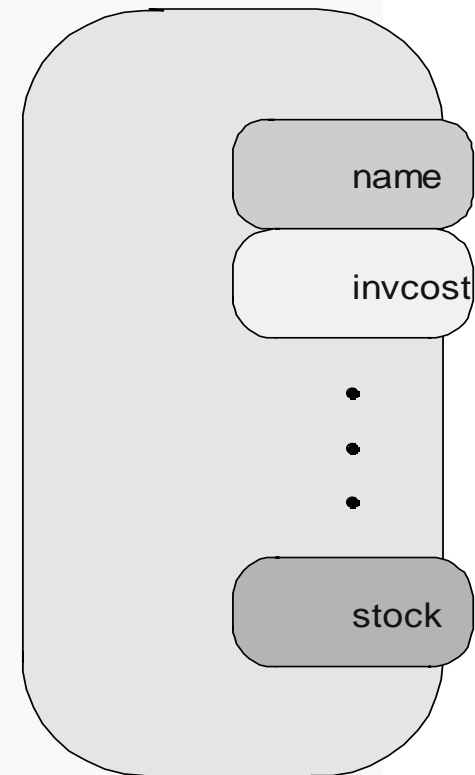
energyRt: main classes

- *technology* – stores technology parameters
- *commodity* – stores commodity information
- *supply* – stores info on resources supply
- *demand* – stores info about final demand
- *constrain* – constrain on endogenous variables
- *repository* – stores commodities, technologies, supply, demand
- *model* – stores main model parameters and all info above
- *scenario* – stores all info above and results from the model run

Adding data to “technology” object, final code

```
ECOA <- newTechnology(  
  name = "ECOA",  
  description = "Coal-fired power plant",  
  input = list(comm = "COA",  
               unit = "PJ",  
               combustion = 1),  
  output = list(comm = "ELC",  
               unit = "PJ"),  
  units = list(capacity = "GW",  
               activity = "PJ",  
               varom = "MRMB/PJ",  
               fixom = "MRMB/GW",  
               invcost = "MRMB/GW"),  
  cap2act = 31.536,  
  afa = list(afa.up = 0.6),  
  ceff = list(comm = "COA",
```

Class 'techno



energyRt: quick reports for objects *model* and *scenario*

A quick report for scenario "scen.BAU"

```
report(scen.BAU)
```

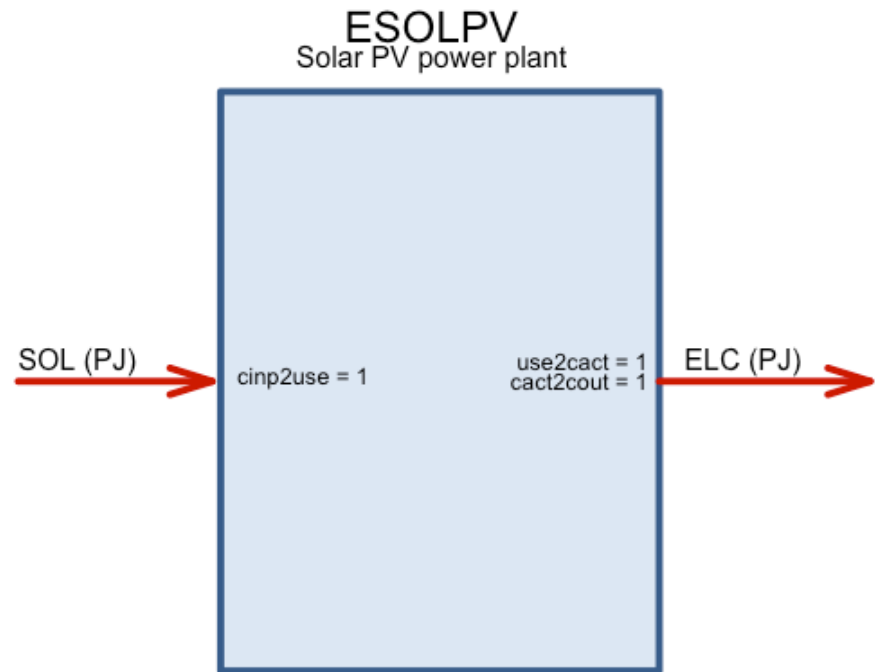


September 16, 2016

1	Optimization	1
1	Summary	1
2	Cost analysis	1
3	Commodity analysis	5
3.1	CCSCO2	5
3.2	CO2	9
3.3	COA	13
3.4	COA1	17
3.5	COA2	17
3.6	ELC	17
3.7	FIELD	20
3.8	GHG	23

energyRt: current functionality

- Analysis of technologies:
 - Design flexible tech. processes
 - Visualize
 - Analysis of levelized costs
 - Comparison
- Design basic Bottom-Up (RES) models
- Running the models and analyze scenarios



energyRt development plan

- Coming soon
 - Time slices (not yet fully implemented)
 - Regions (not yet tested well)
- Mid-term
 - Check of inconsistencies in data, general bugs
 - Visualizations tools, including:
 - Geographic Information System (GIS)
 - Sankey plots (energy or any commodities flows)
 - Automatic units tracking and conversion
- Long term
 - More models included (Integrated assessment set)

energyRt

energy systems modeling R-toolbox

[View on GitHub](#)

[Download .zip](#)

[Download .tar.gz](#)

Thank you for your attention!

<http://energyRt.org>

Welcome to energyRt Pages

energyRt is a package for [R](#) to develop Reference Energy System (RES) models and analyze energy-technologies.

energyRt package includes a standard RES (or "Bottom-Up") linear, cost-minimizing model, which can be solved by [GAMS](#) or [GLPK](#). The model has similarities with [TIMES/MARKAL](#), [OSeMOSYS](#), but has its own specifics, f.i. definition of technologies.