UBA
Universidad de Buenos Aires

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Overview of the Modelica-based System Dynamics Library

Prof. Dr. Rodrigo Castro
rcastro@dc.uba.ar

MODELING OF SUSTAINABLE ECONOMIC SYSTEMS

MOSES 2016
May 16-20
Linköping, Sweden

LINKÖPING UNIVERSITY

- This lecture borrows several pieces of content from presentations on global modeling and the Modelica-based library for System Dynamics kindly made available by Prof. Dr. François E. Cellier (emeritus), ETH Zurich

The System Dynamics methodology

- Introduced in the late sixties by J.W.Forrester
  - A tool for visually organizing partial knowledge about models of poorly understood systems in phenomenological sciences
    - Biology, Ecology, Macro economy, Sociology, etc.
- Low-level modeling paradigm: “Stocks and Flows”
  - **Stock** elements connected by material or non-material (e.g., information) **Flows**
  - **Flows** regulated by input-output **Rates**
  - A simplified way of dealing with differential equations
    - Shuns traditional calculus
- In this talk we shall learn about an SD library built with the Modelica language for equation-based modeling, and why it can be of help

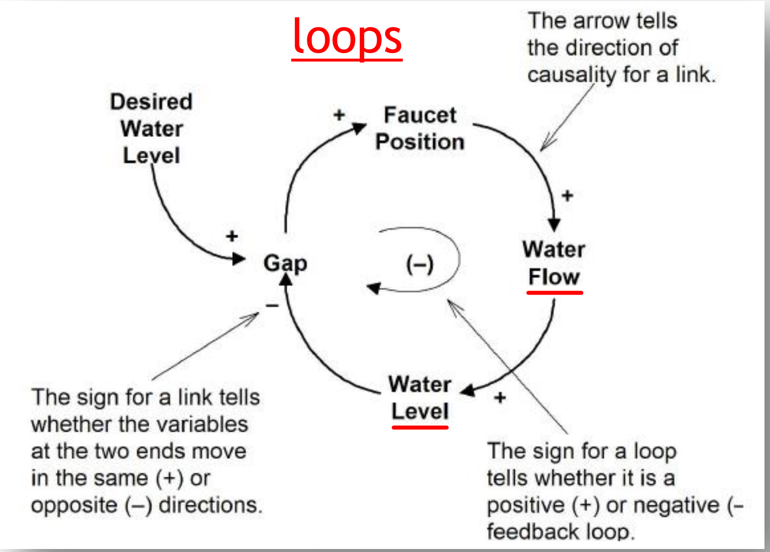
The System Dynamics methodology

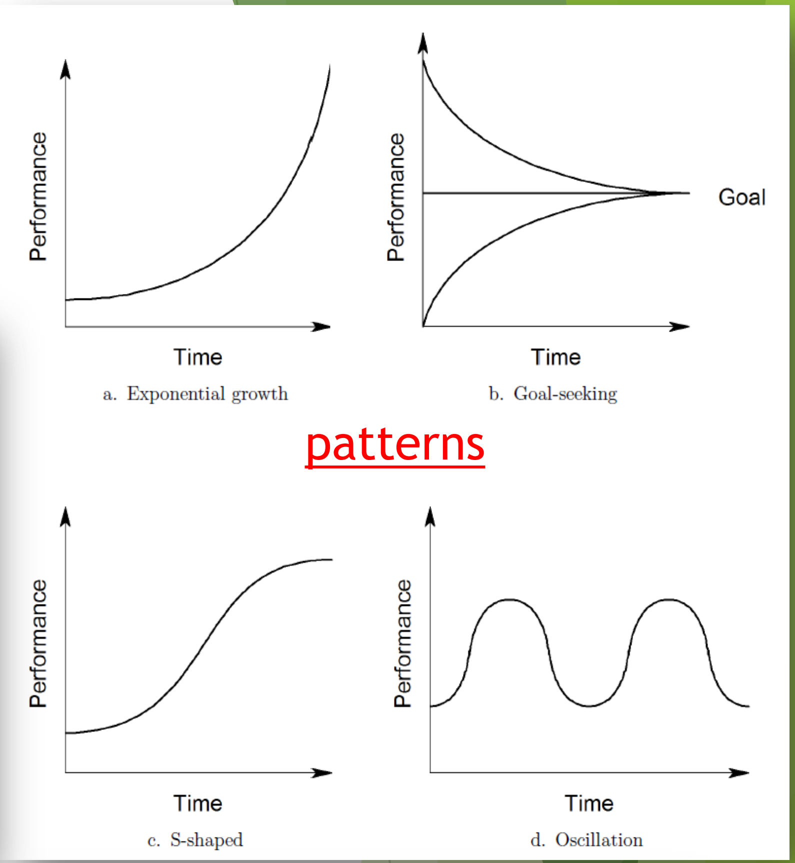
- Physical systems call for **deductive modeling**
  - Well established meta laws exist. These are preserved:
    - across a wide range of spatio-temporal scales
    - under system composition/decomposition procedures
  - Very reliably modeled
  - *Example: a car (mechanical-electro-computerized system)*
- Ill-defined systems call for **inductive modeling**
  - Much more complex to model in a reliable way
  - Very weak, narrowly applicable, or totally inexistent meta laws
  - Difficult decomposability into subsystems (densely connected)
    - Submodel parameters are influenced by many system's variables in similar orders of magnitude
  - *Example: an ecosystem (bio-geo-chemical system)*
- System Dynamics
  - Applicable in both deductive and inductive modeling
  - Useful only when used carefully (avoid to incur in a "modern reductionism")

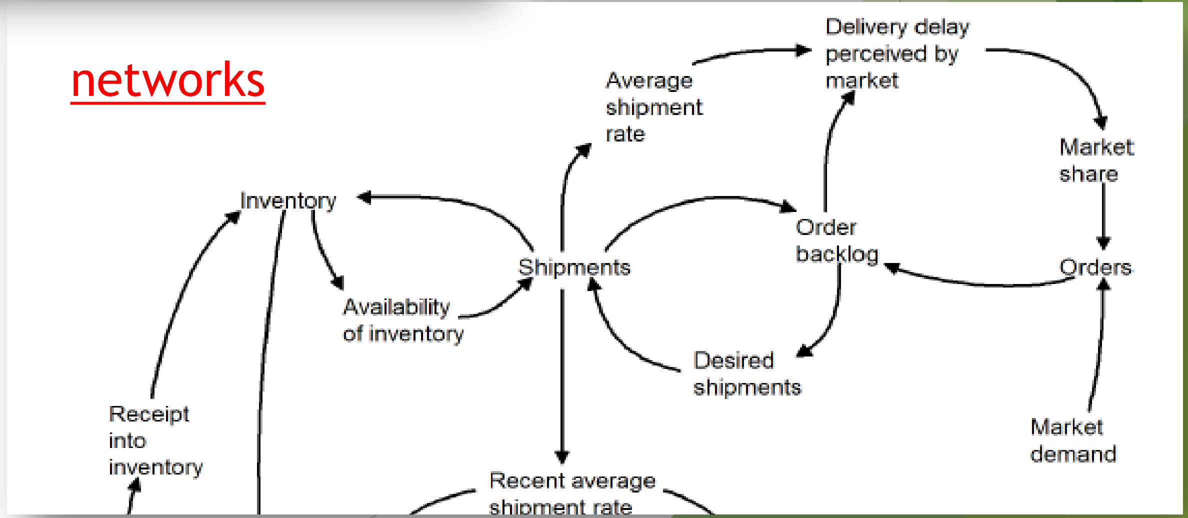
The System Dynamics methodology

- Explore dynamic behavior of systems lacking universal laws
- Think visually in terms of:
  - Basic system structures as positive and negative feedback loops
  - Patterns of behavior
- Loops forming complex interdependent networks
  - Opposed to simplistic independent unidirectional cause-effect relations
- Also accounts for:
  - Time delays in internal system flows
  - Nonlinear effects

from System Dynamics Methods: A Quick Introduction, C. Kirkwood



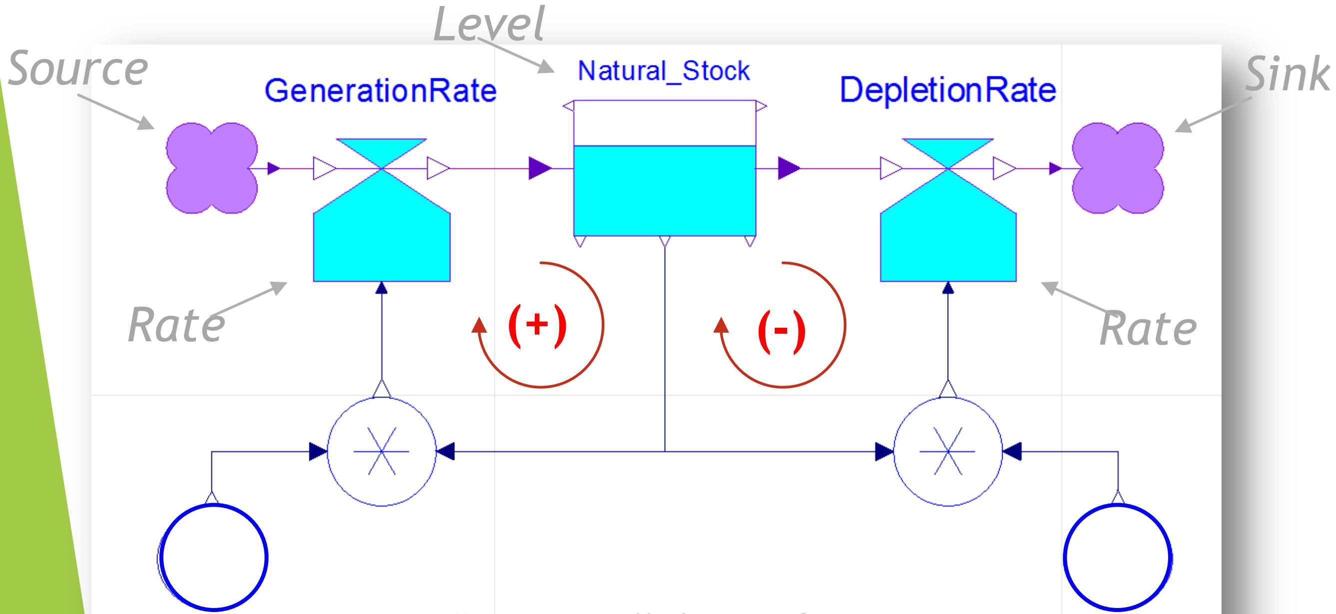




Simple exponential generation/depletion model

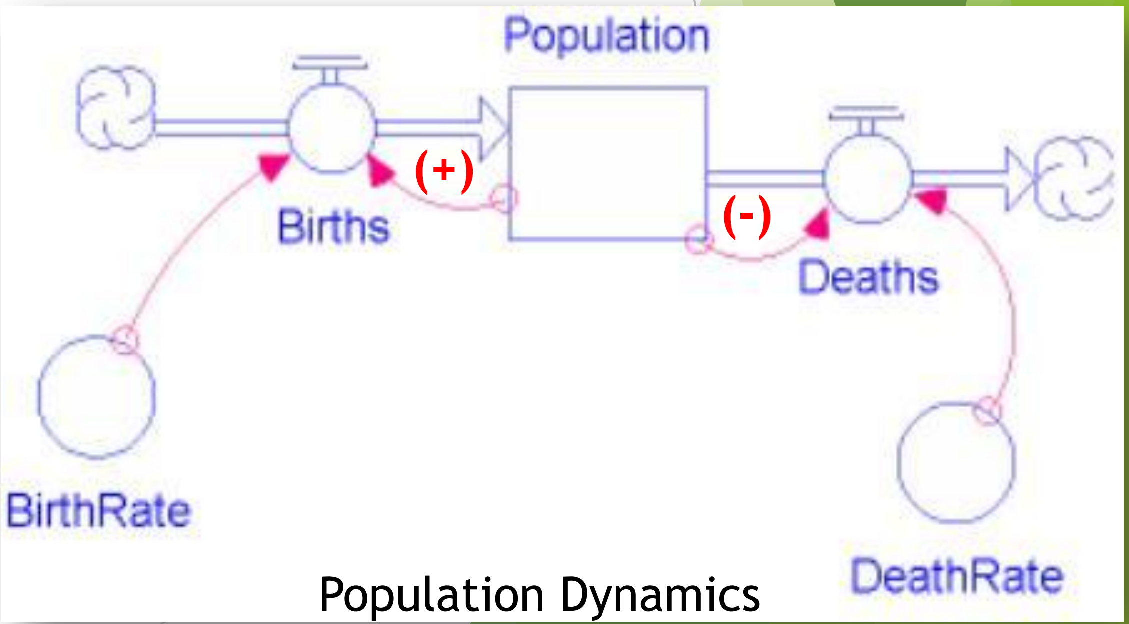
- Example with two loops: 1 negative, 1 positive
- **Levels** and **Rates**
- **Sources** and **Sinks**
  - Provided for optical purposes only (System Dynamics modelers are used to them)
    - These models do not represent equations.

E.g.: The Modelica-based SD Library + OpenModelica Tool



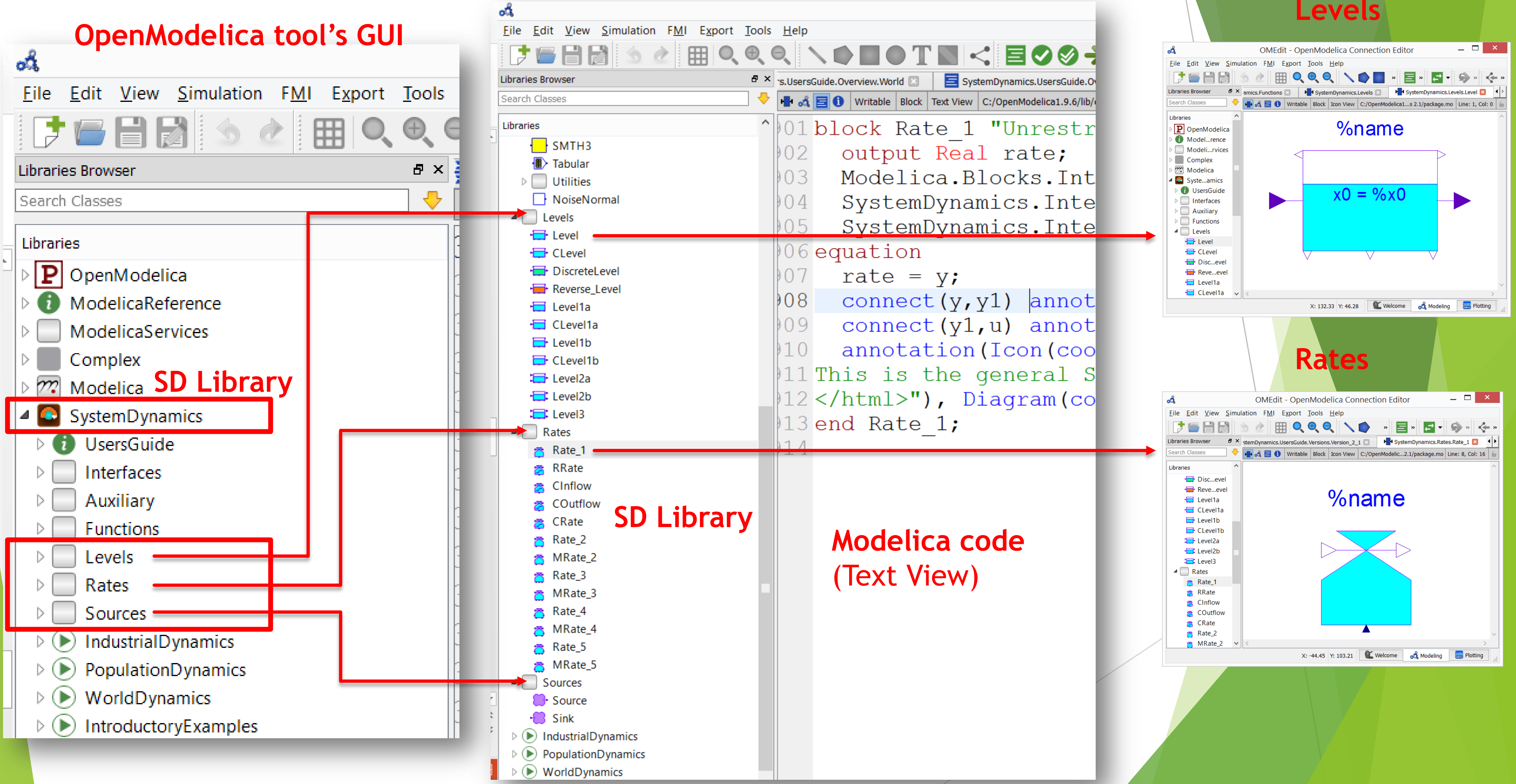
A “generic” formulation

E.g. The Stella SD-based Tool



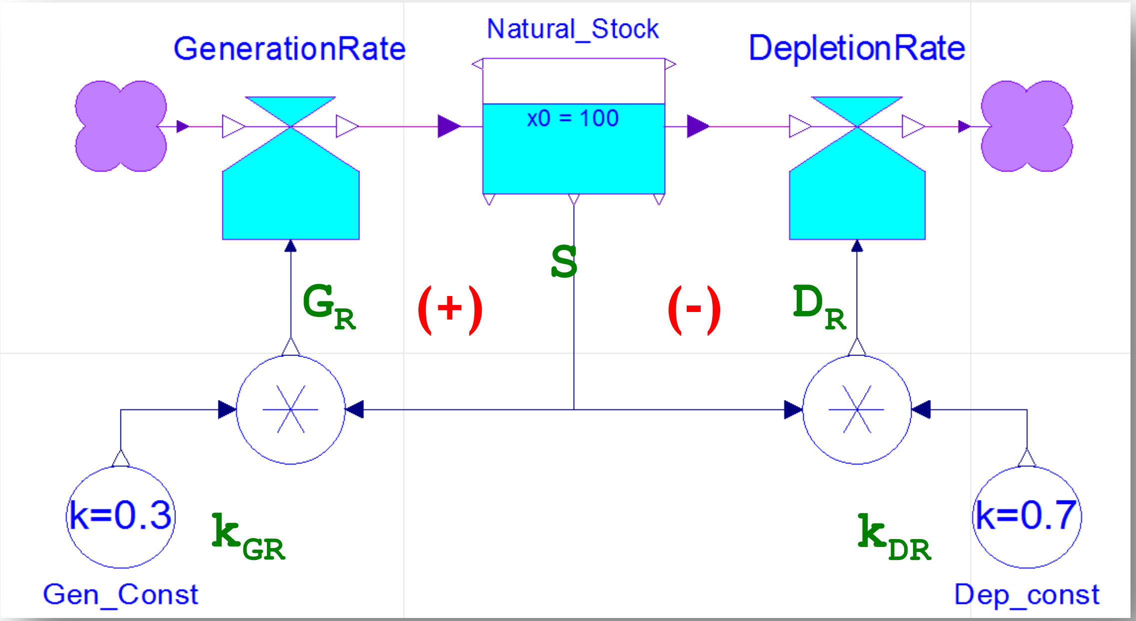
Population Dynamics

Mod-SD: The SystemDynamics library for Modelica



Simple exponential generation/depletion model

- Required model parameters:
  - Level: initial condition
  - Rates: constant coefficients
- Required simulation parameters:
  - Initial and final simulation time
  - Numerical accuracy desired



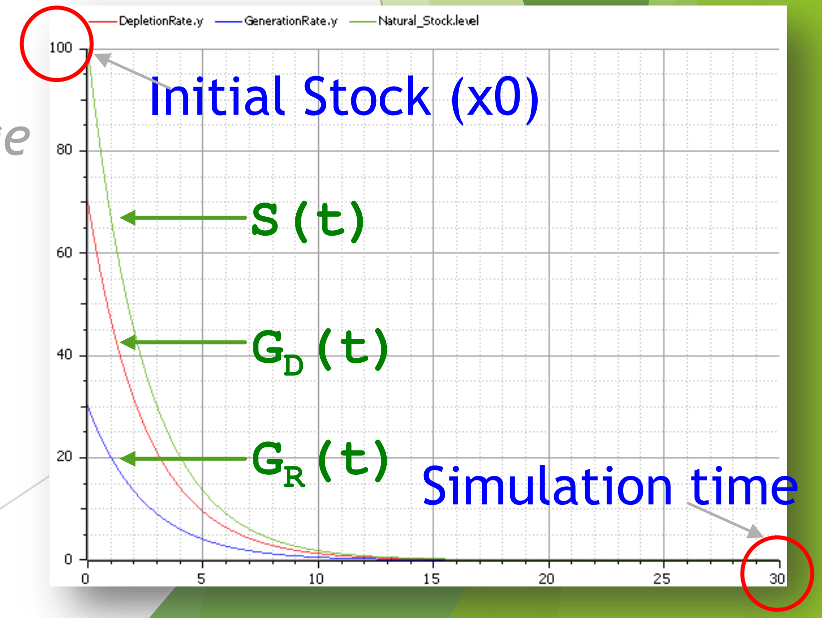
Rate of Level change

$$\dot{S} = + G_R - D_R$$

$$S(t=0) = x0$$

$$G_R = k_{GR} \cdot S$$

$$D_R = k_{DR} \cdot S$$

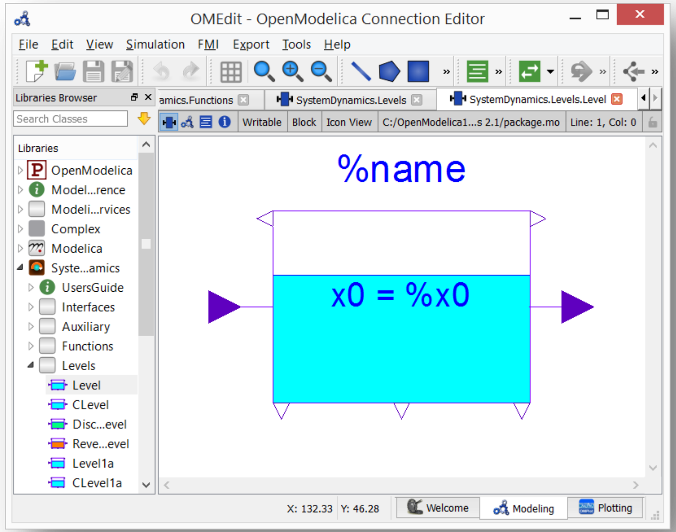


Mod-SD: Levels

Levels represent the **state variables** of the System Dynamics modeling methodology.

$$\dot{y} = + u1 - u2$$

Icon View



Text View

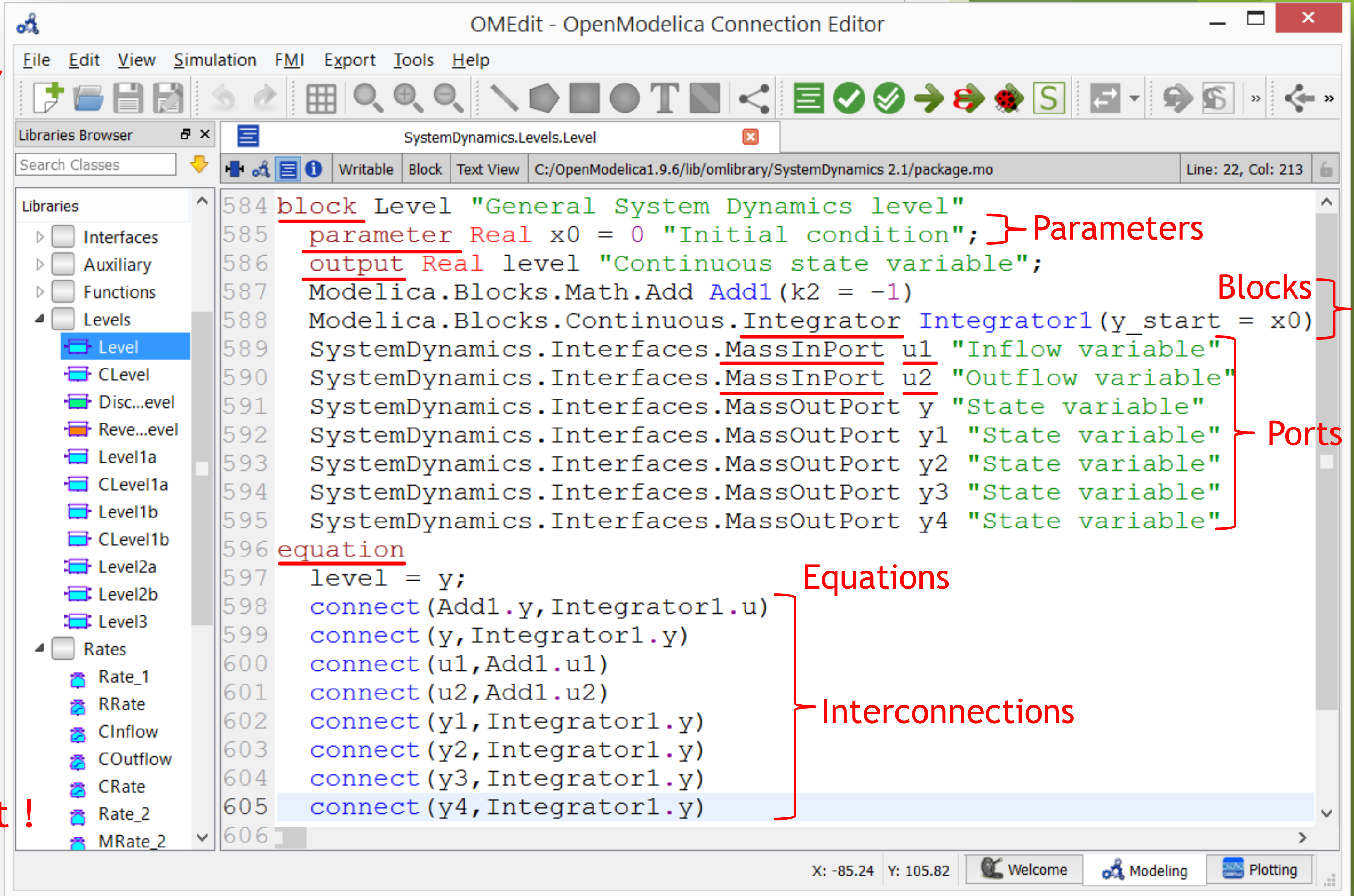
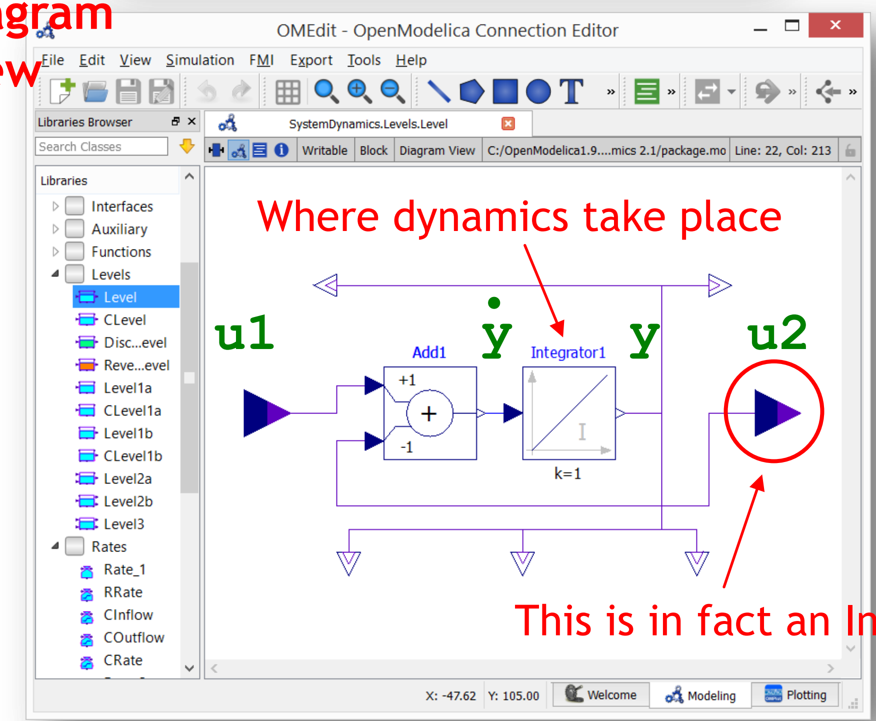


Diagram View



Mod-SD: Levels

Icon View

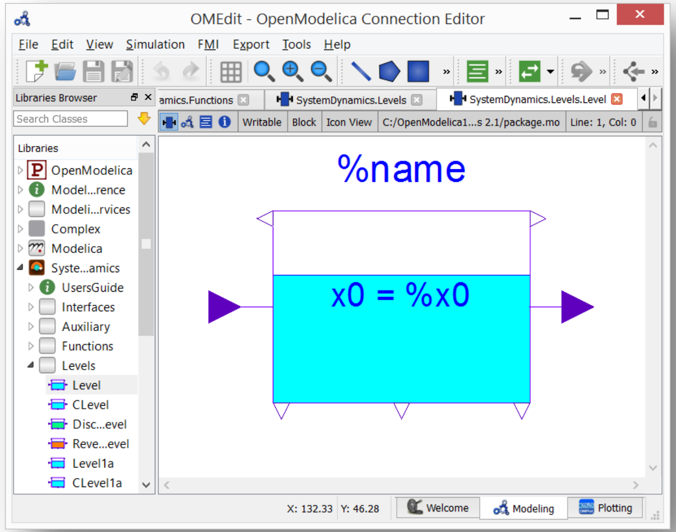
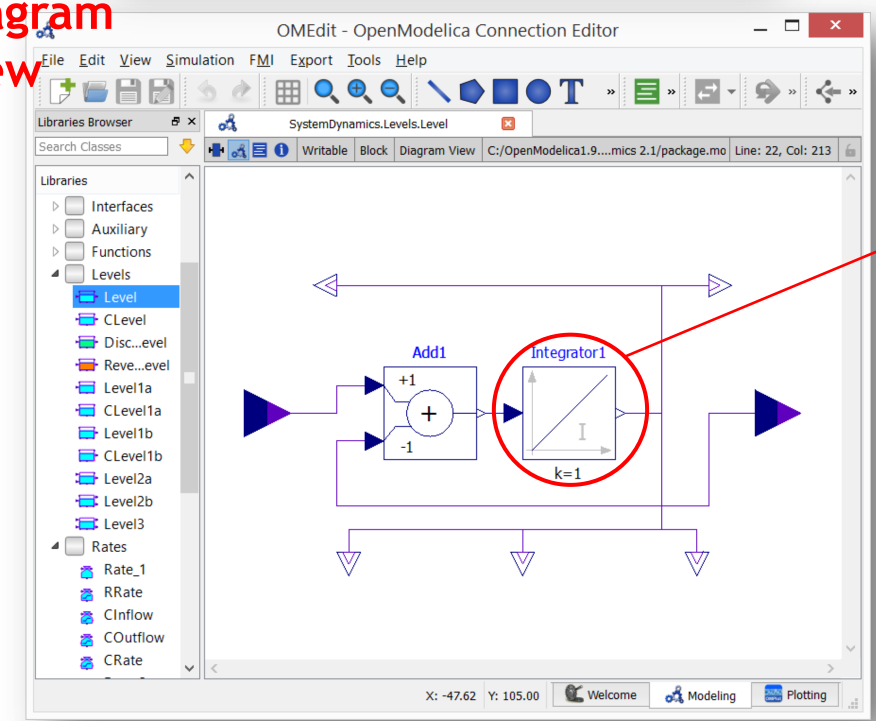


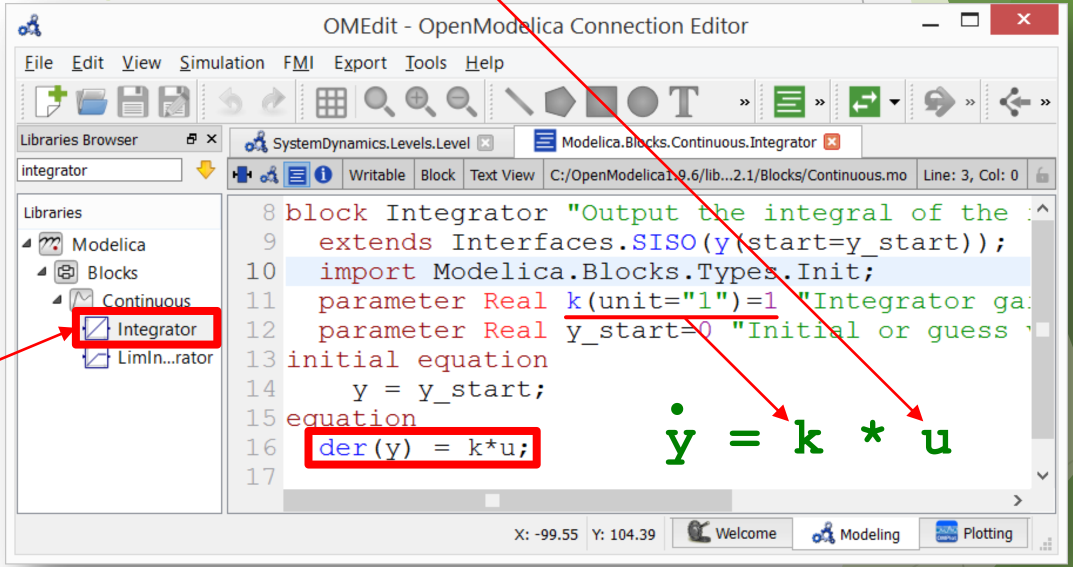
Diagram View



Levels represent the **state variables** of the System Dynamics modeling methodology.

$$\dot{y} = + u1 - u2$$

Integrator block

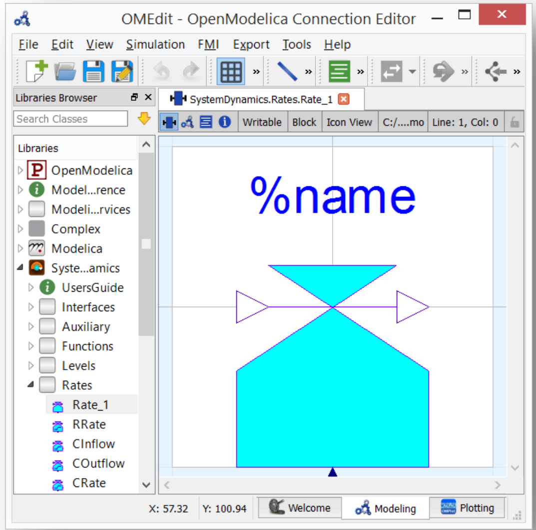


$$\dot{y} = k * u$$

- The integration operation is defined in an independent standard block
- It is not required to decide on the time step for the numerical solution
- Model is separated from simulation

Mod-SD: Rates

Icon View



Rates define the **state derivatives** of the System Dynamics modeling methodology.

Text View

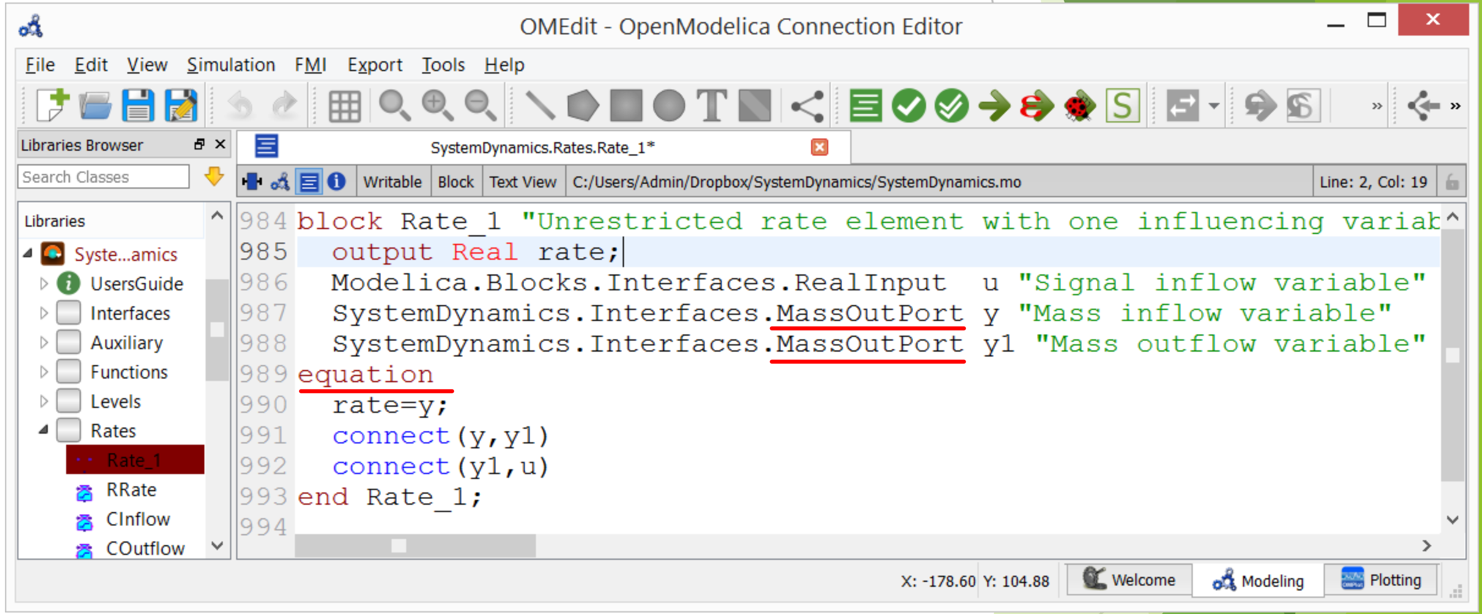
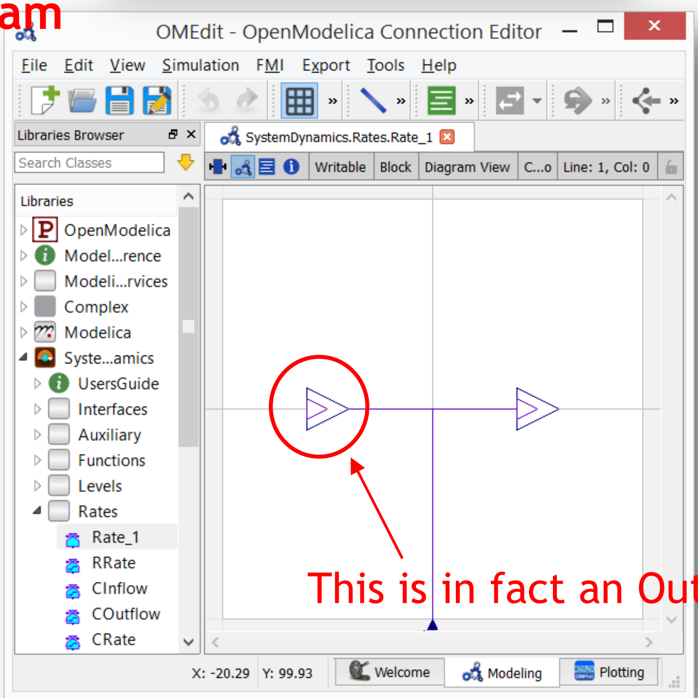


Diagram View

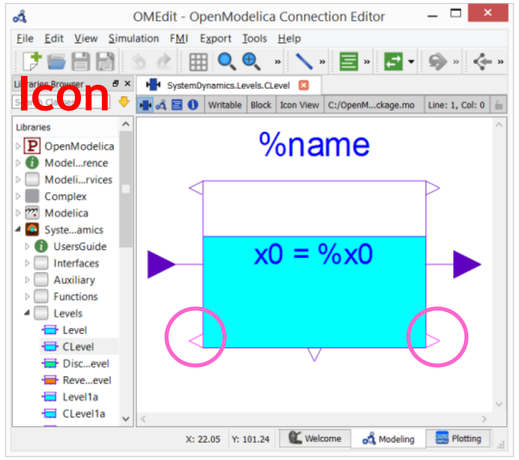


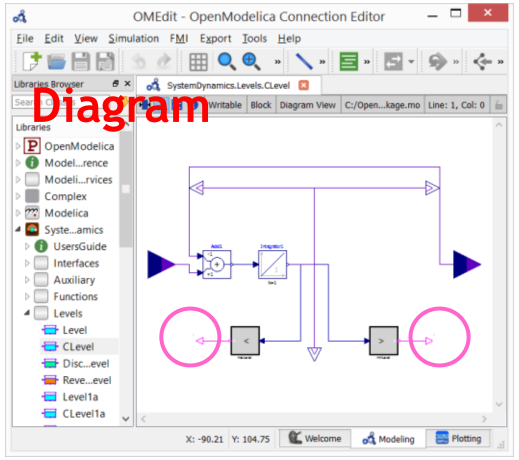
This is in fact an OutPort !

- No "dynamics" defined in this model
- Only a wiring of information delivered to other components connected to it

Mod-SD: Extended convenience models

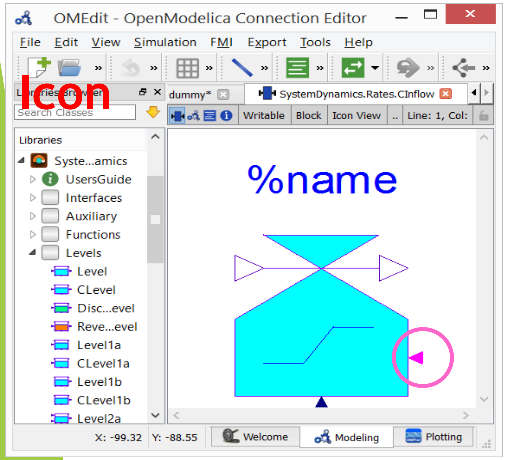
Controlling Level

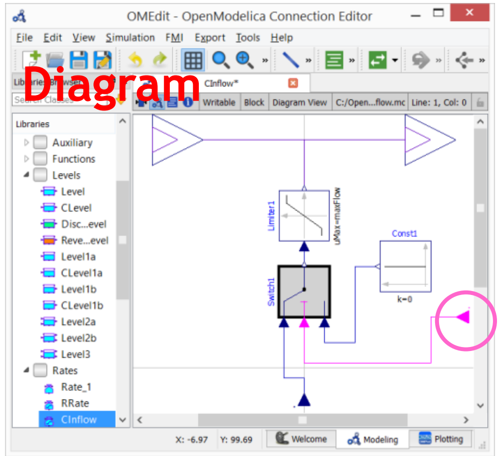




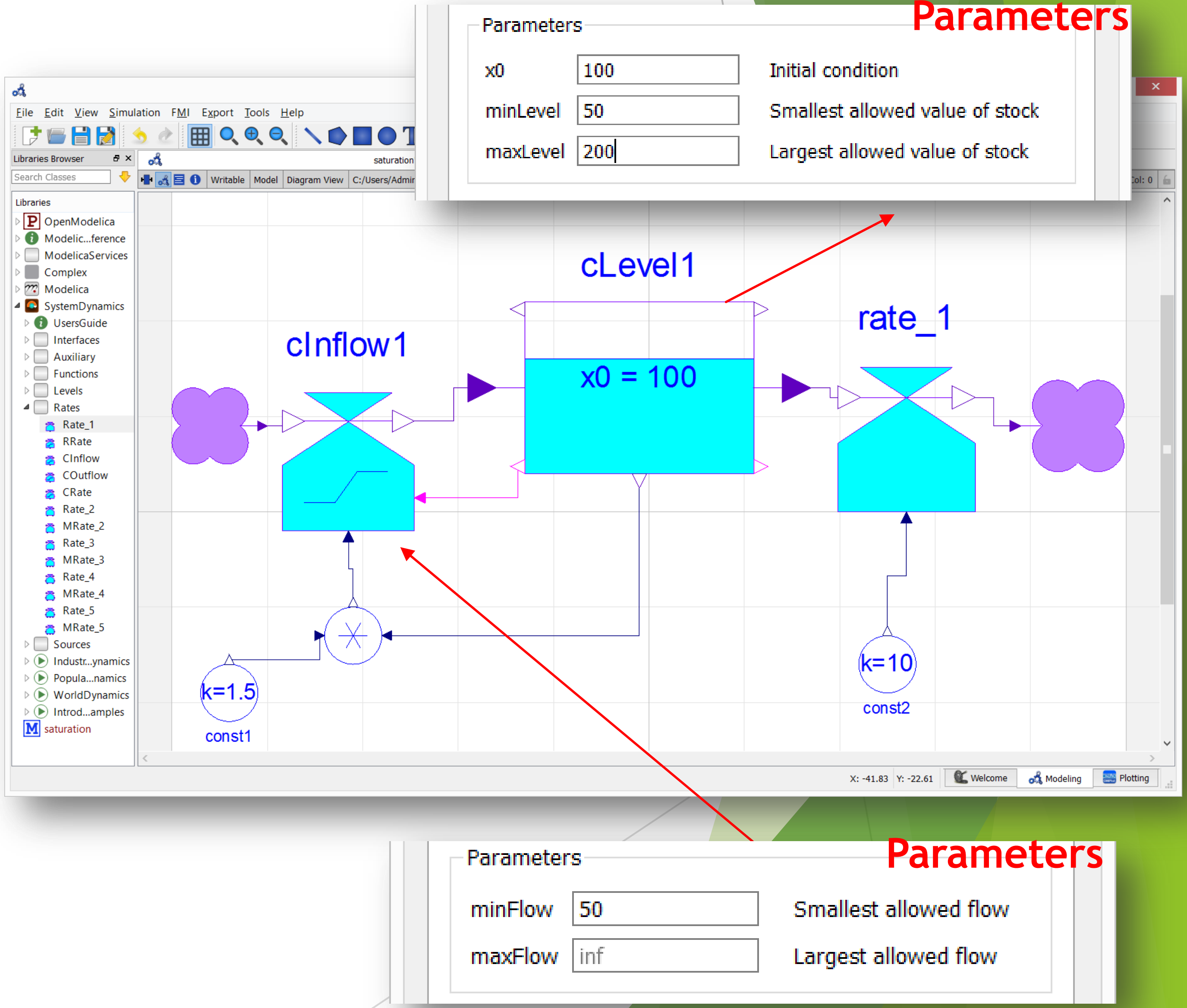
Outputs logical information on whether minimum or maximum thresholds are crossed

Controlled Inflow Rate, *with saturation*



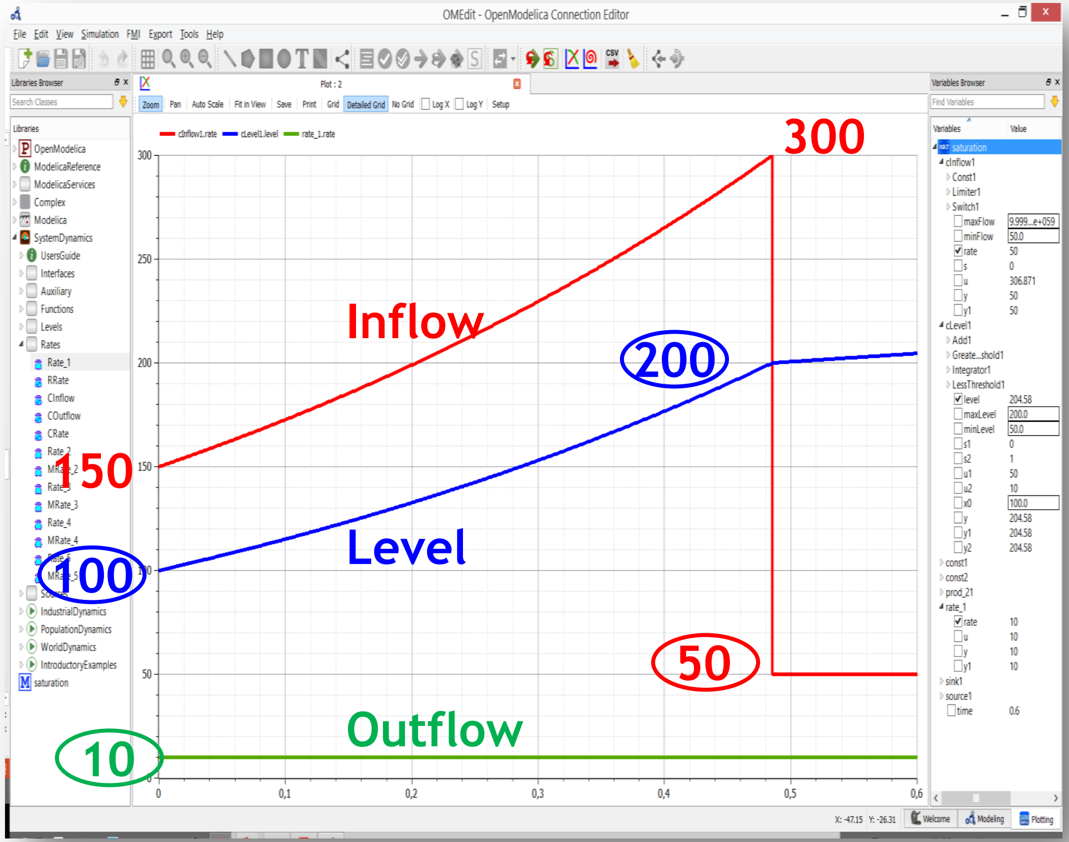


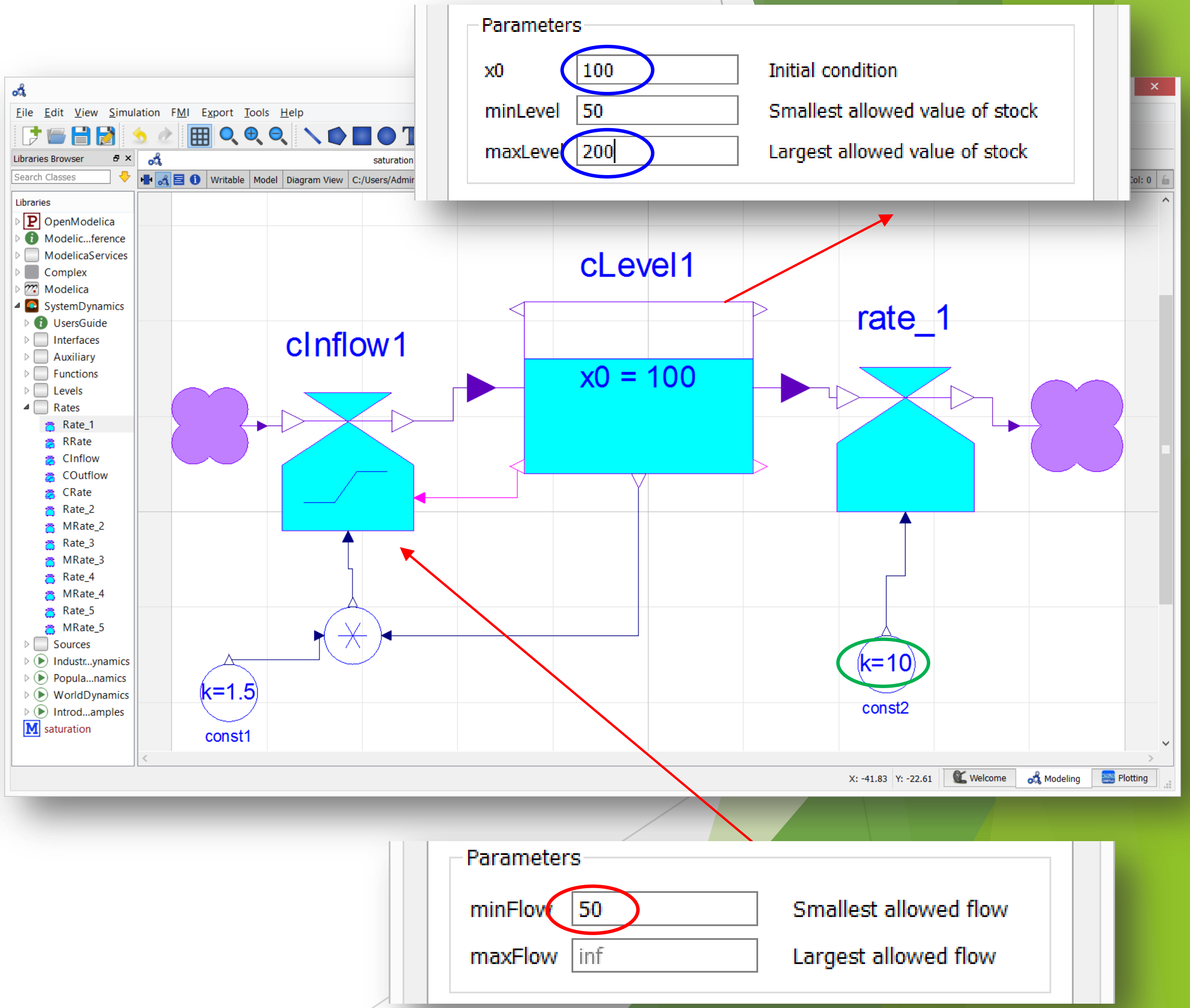
Saturates at certain lower and upper values, and can switch to a constant if commanded so



Mod-SD: Extended convenience models

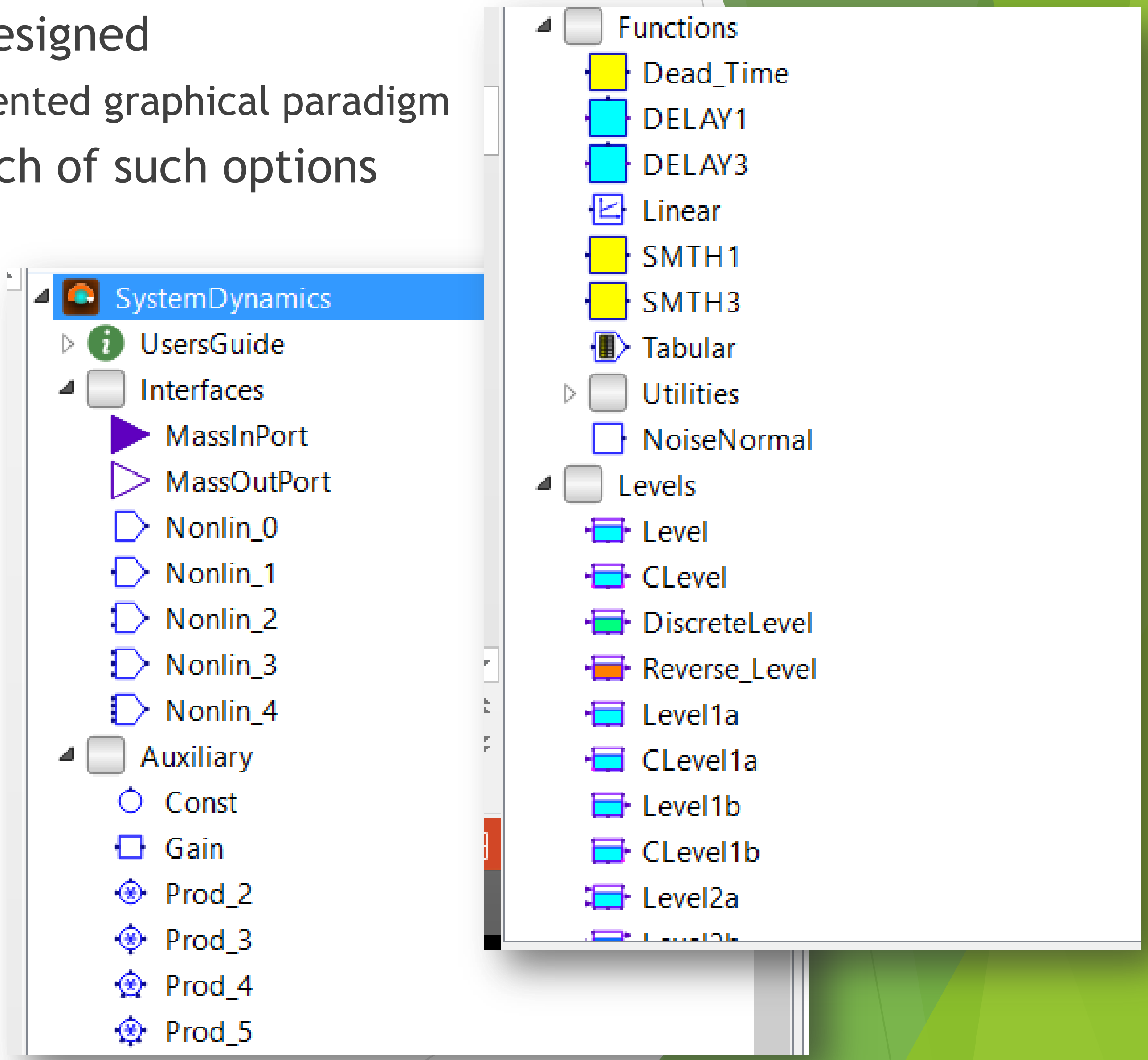
- Example results



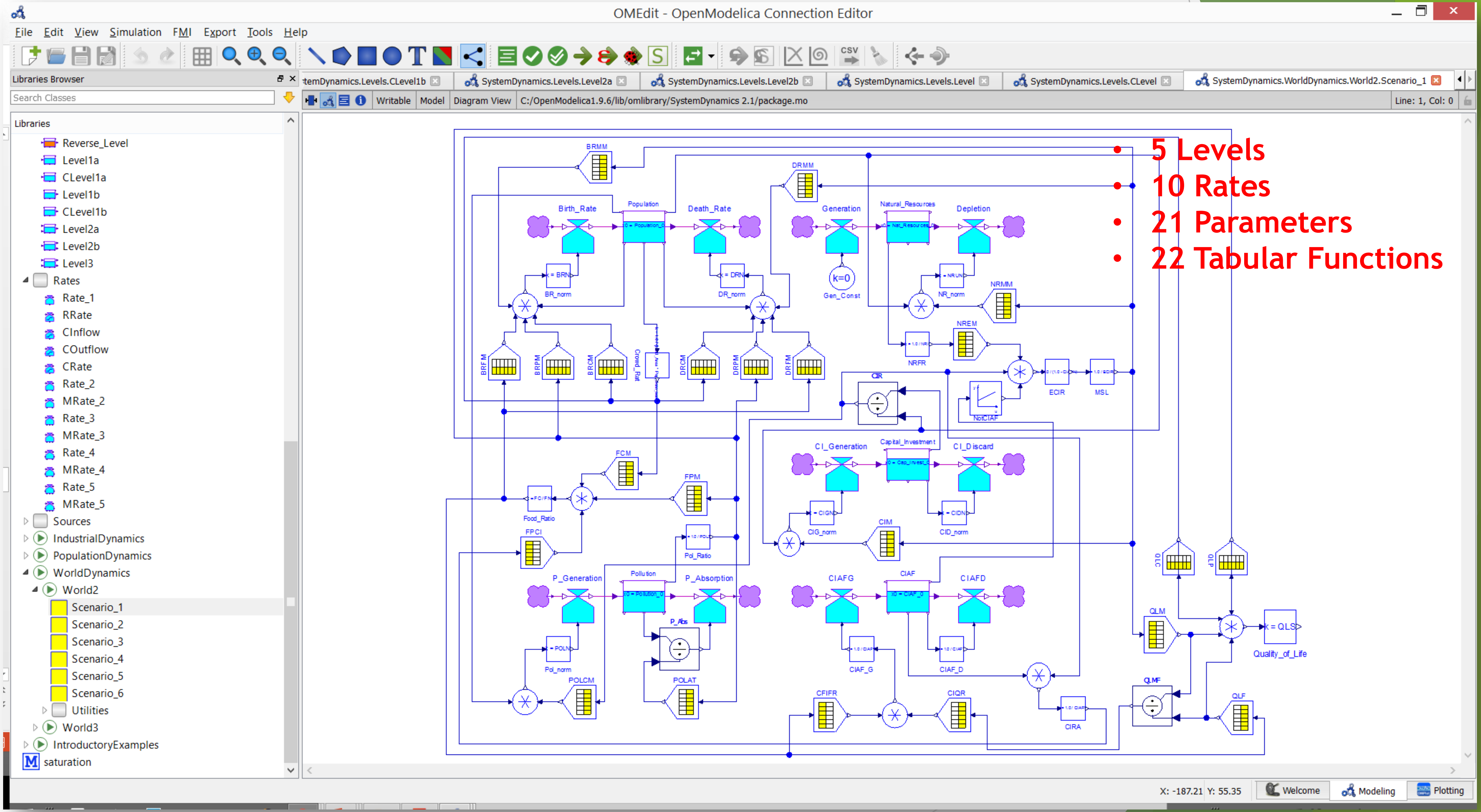


Mod-SD: Extended convenience models

- New advanced/handy basic elements can be designed
  - Relying on the Modelica language and its block-oriented graphical paradigm
- The standard Mod-SD library comes with a bunch of such options
  - Rates
    - Controlled/Saturated Rates
    - Multiplicative Rate with several inputs
    - Additive Rate with several inputs
  - Levels
    - Controlled/Saturated Levels
    - Discrete time Level
    - Reverse time Level
    - Multiple inputs/Multiple outputs Level
  - Interfaces, Auxiliary, Functions
    - Delays
    - Dead Time
    - Smoothers
    - Tabular, Linear and Non Linear Functions
    - Gains, Constants, Multi-input products

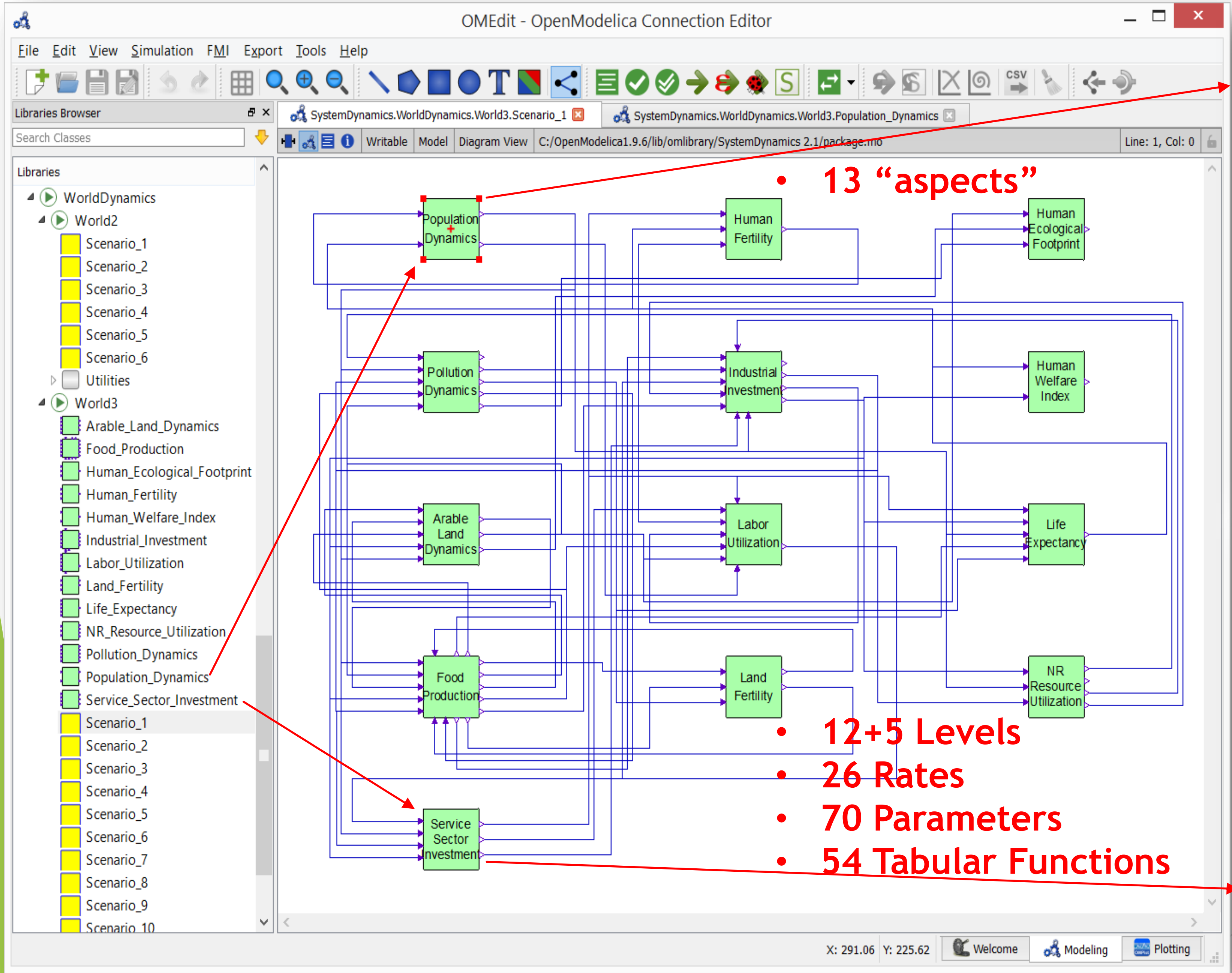


A simple World Model with Mod-SD: World2



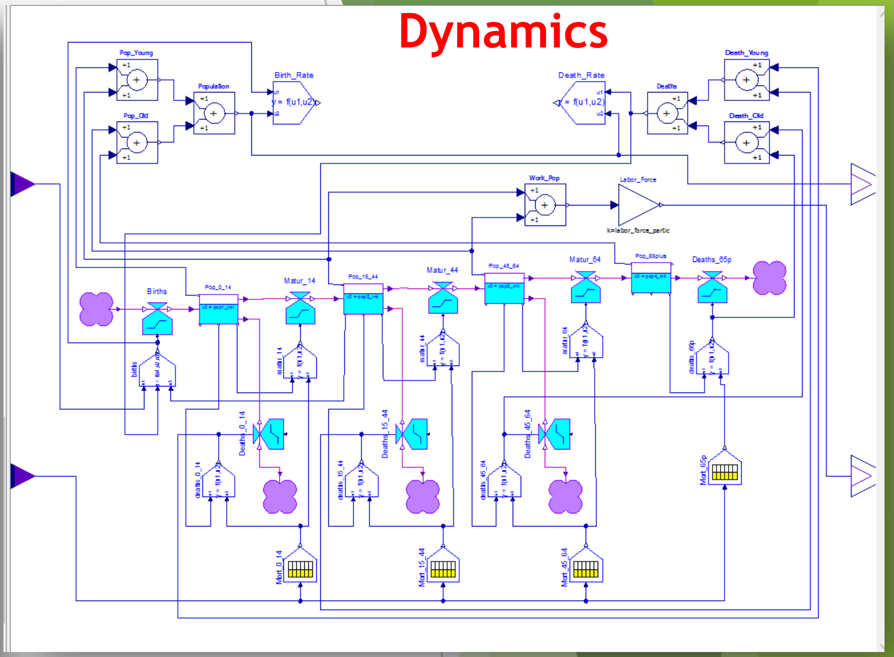
- 5 Levels
- 10 Rates
- 21 Parameters
- 22 Tabular Functions

A complex World Model with Mod-SD: World3

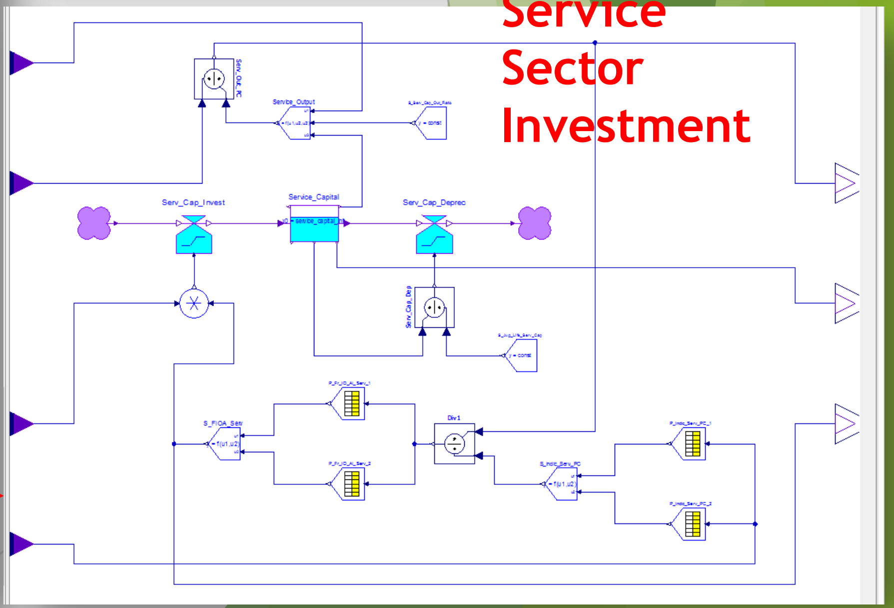


- 13 “aspects”
- 12+5 Levels
- 26 Rates
- 70 Parameters
- 54 Tabular Functions

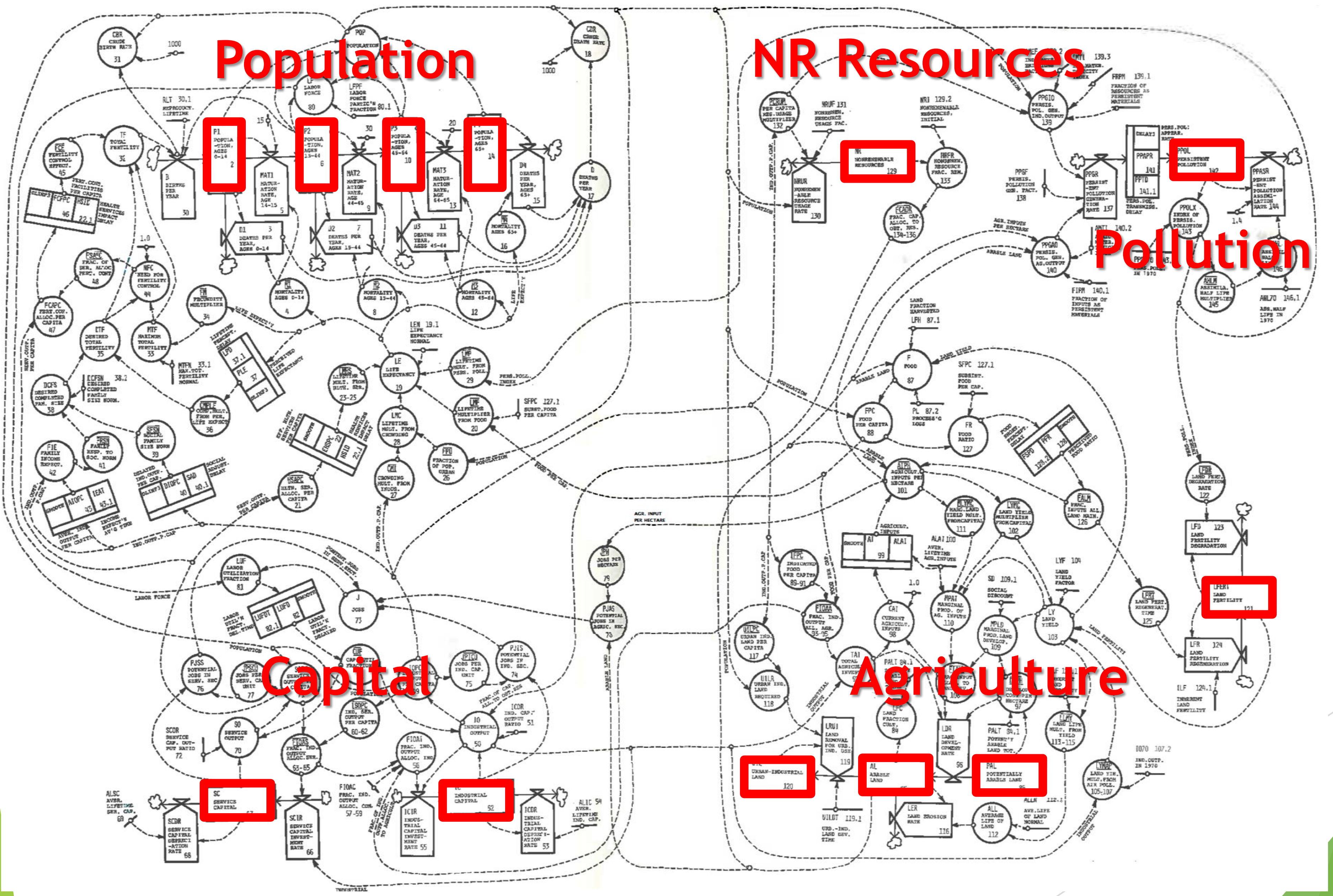
Population Dynamics



Service Sector Investment

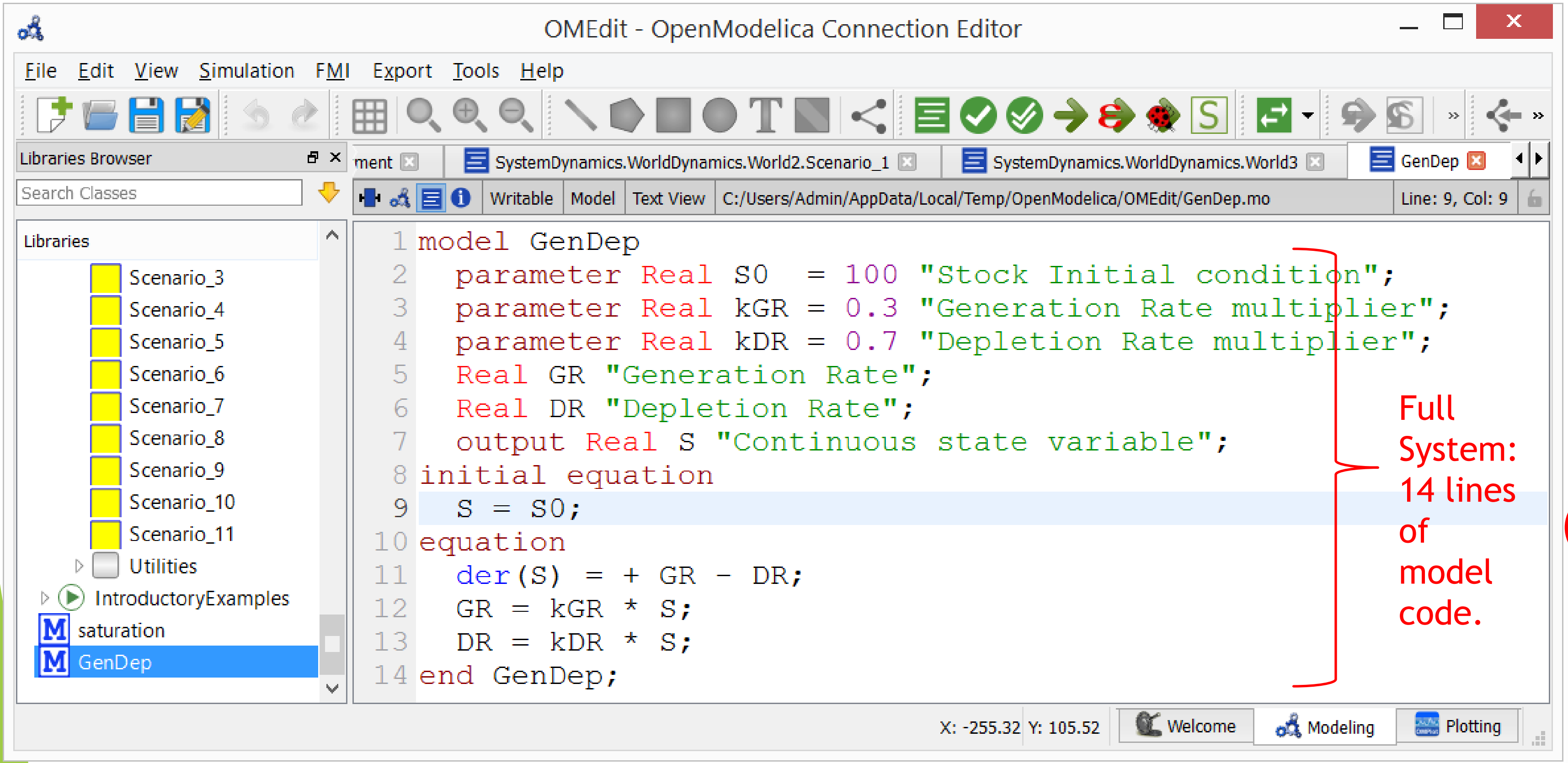


A complex World Model with SD: World3



Simple exponential generation/depletion model (again)

- “Pure Modelica” code required for our simple growth model

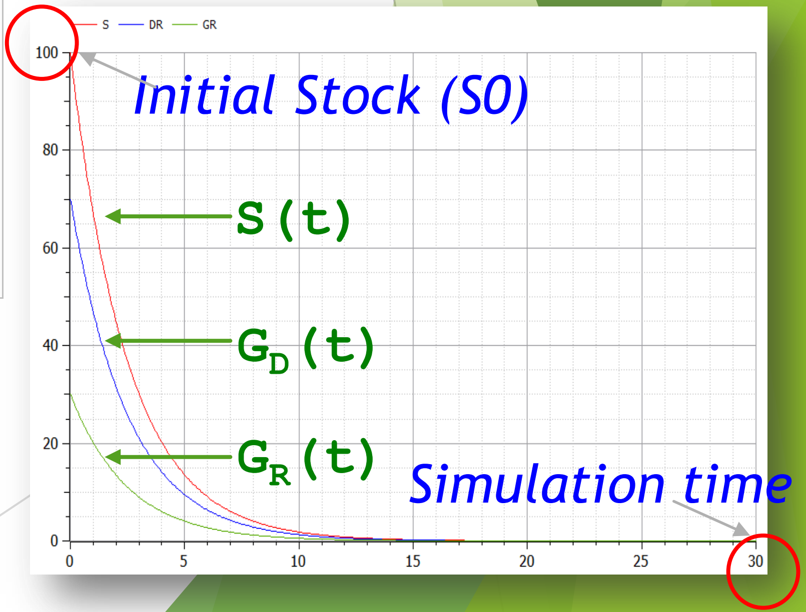


$$\dot{S} = + G_R - D_R$$

$$S(t=0) = S0$$

$$G_R = k_{GR} \cdot S$$

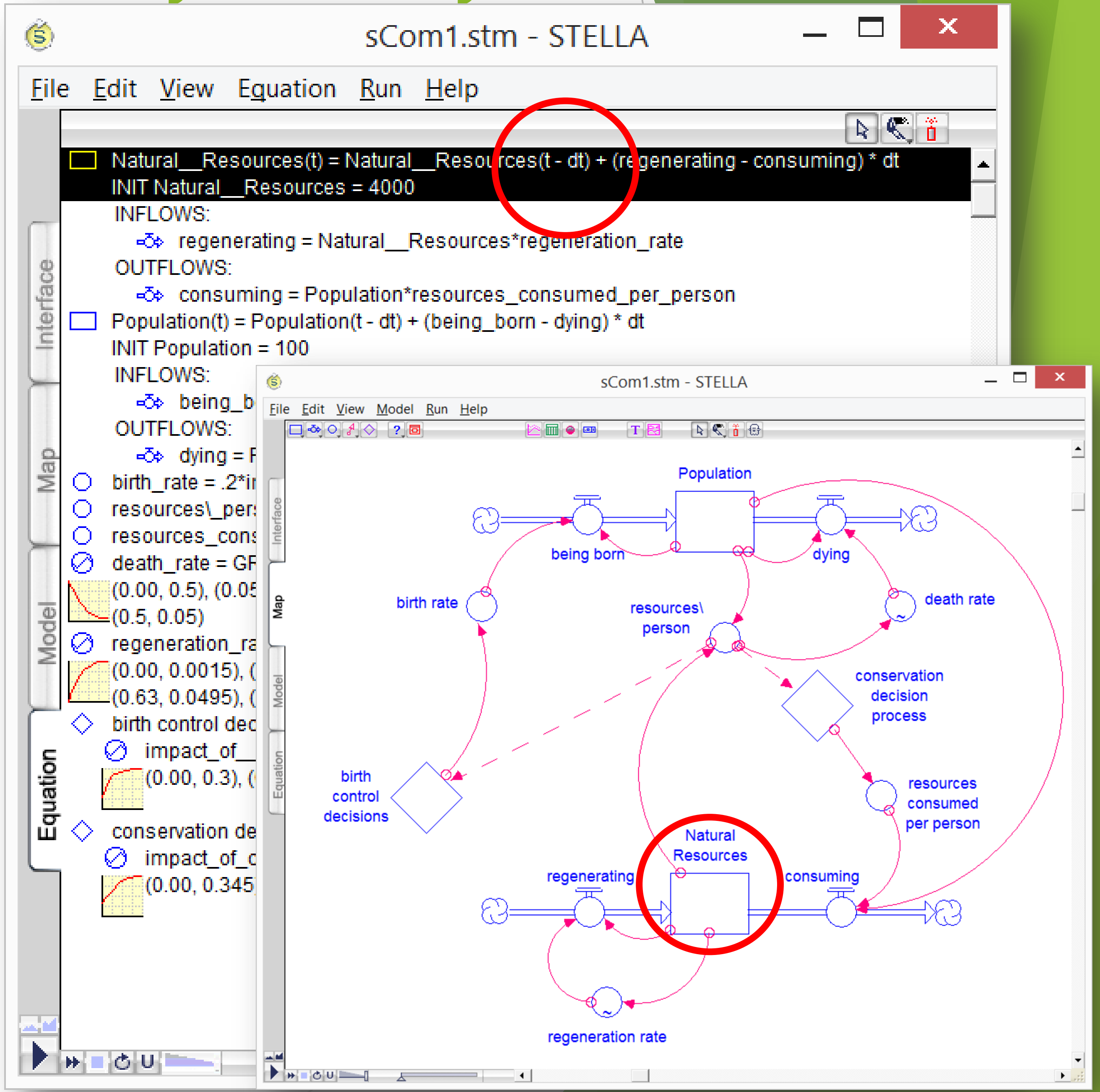
$$D_R = k_{DR} \cdot S$$



- Much more compact ! Exactly the same results
- So… why Mod-SD ? What do we gain ?

The tool-centered approach to System Dynamics

- No separation between GUI, Model and Simulation
- At the very heart of System Dynamics there is a tight bind between the models and the numerical simulation aspects
- Textual model specifications are not standardized
  - Different tools lead to different code



A Modelica-based System Dynamics Library

- System Dynamics is a fairly low-level modeling paradigm
  - Its implementation does not place heavy demands on the modeling software
  - Modelica may be an overkill for dealing with System Dynamics models
  - **Levels**, **Rates** and **Transformations** are the core of System Dynamics
    - Are so simple that their implementation in Modelica requires very little time and effort
- The value of the **Mod-SD** library is not in its basic models, but rather in the resulting standard and open application codes
- Methodologically it offers a sound bridge between **deductive and inductive modeling**
  - Combination of System Dynamics models with the vast object-oriented, multi-formalism, cyber-physical modeling capabilities of Modelica
- Sound integrated treatment of heterogeneous socio-technical systems
  - Continuous-time, discrete-time, and discrete-event aspects

A Modelica-based System Dynamics Library

- Key issues: **scalability, flexibility, modularity, robustness**
  - For entry level systems we can be well off with almost any tool for SD
  - When the complexity of systems grows (dynamics/structure/size) we need scale up safely and flexibly: we need **more robust tools**
    - A **MUST** in serious interdisciplinary global modeling. E.g. the MOSES collaboration
- Modelica is currently the most advanced technology for equation-based systems modeling
- Mod-SD can accommodate flexibly different levels of expertise:
  - High level modeling: Graphical reasoning on complex systems (out-of-the-box “usual SD”)
  - Low level modeling: New/advanced models, structures (“extended SD”)
  - Advanced simulation: performance, optimization, sensitivity analysis, etc.

Conclusions

- Based on the Modelica ecosystem of technologies, we can leverage the pre existing knowledge base o System Dynamics to cope with the requirements of the next generation of global models
- System dynamics was introduced as a methodology that allows us to formulate and capture partial knowledge about any soft-science application, knowledge that can be refined as more information becomes available
- Systems dynamics is the most widely used modeling methodology in all of soft sciences. Tens of thousands of scientists have embraced and used this methodology in their modeling endeavors

Questions

?

Contact: rcastro@dc.uba.ar http://dc.uba.ar/People/rcastro