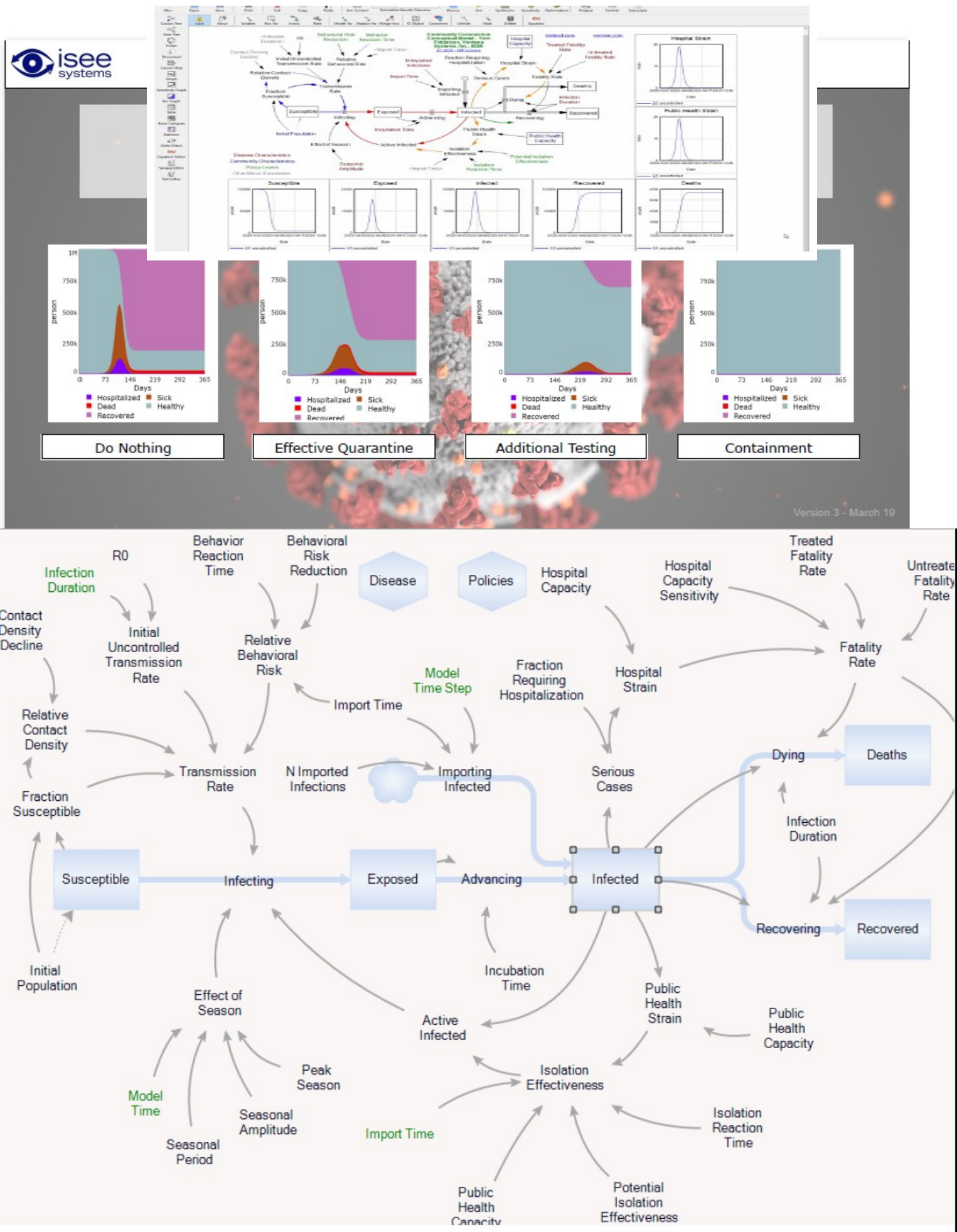


Lecture 21: SYSTEMS DYNAMICS -- LOTKE-VOLTERRA PREDATOR PREY MODELS AND MORE



Systems dynamics is a tool for simulating links between agents by **differential/difference equations**. The models exploit non-linearities and feedback loops to capture the dynamic relations in the “actual process”. (http://en.wikipedia.org/wiki/List_of_system_dynamics_software).

Systems Dynamics was developed by Jay Forrester an MIT engineer who invented flight simulators to give social sciences a systems perspective. Forrester wrote Urban Dynamics, and World Dynamics. In 1960s the Club of Rome, a group of businessfolk/ intellectuals interested in the future of the world publicized Dennis Meadows’ et al Limits of Growth which NYT columnist Antony Lewis called “one of the most important documents of our age”. http://en.wikipedia.org/wiki/Jay_Wright_Forrester .

How did the systems dynamicist determine their model of the world? Forrester: “... from intensive discussion with a group of people who know the system first hand”. This produced parameters and relations based on no verified information. In 1973 Bill Nordhaus critiqued Forrester’s book “ World Dynamics: Measurement without Data”, noting that not a single relationship or variable was drawn from actual data/empirical studies.

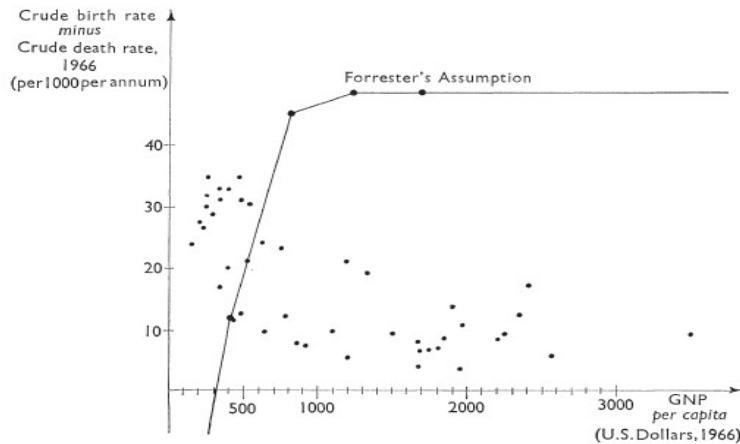


FIG. 2. Assumed and Cross Section Population Growth, 1966.
The line marked “ Forrester’s assumption ” assumes that food and consumption rise in proportion with *per capita* G.N.P.; and that crowding and pollution are at 1970 levels.
Each of the unconnected dots represents one country: the sources of the data are the U.N. *Statistical Yearbook* and U.N. *National Accounts Statistics*, Vol. II.

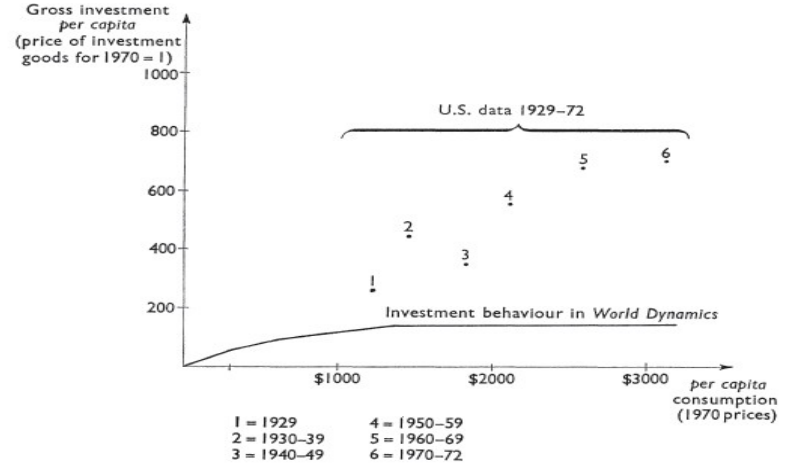


FIG. 5.
Solid line shows implicit investment behaviour in *World Dynamics*. Dots show actual investment behaviour for the U.S. for the period 1929-1972.

The prediction of the systems dynamicists was a Malthusian overpopulation disaster. In 1992 Meadows published a revision of his work entitled Limits of Growth Revisited. Nordhaus critiques this book in Lethal Model 2. Climate change models have similar structure but based on scientific evidence on relations and size of parameters.

Systems Dynamics:

- 1) set up differential/difference equation models to guide thinking about the PROCESS underlying a problem;
- 2) treats $x \rightarrow y$ flow diagrams as differential equation in which $dy = bdx$ and thus turns hand waving “theories” into testable propositions depending on estimable parameter.
- 3) Mock regression models as a LAUNDRY LIST kitchen sink approach that produces descriptive static models based on correlations rather than reflecting the process. STELLA program has the following:

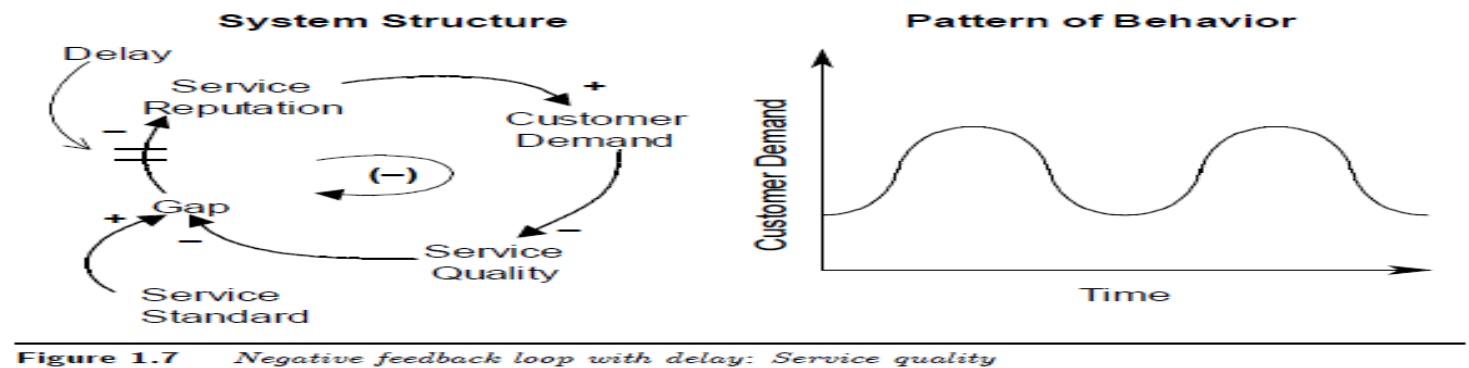
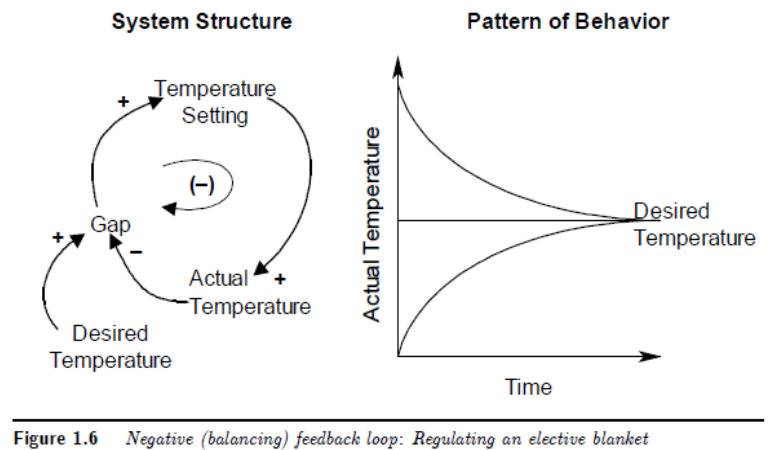
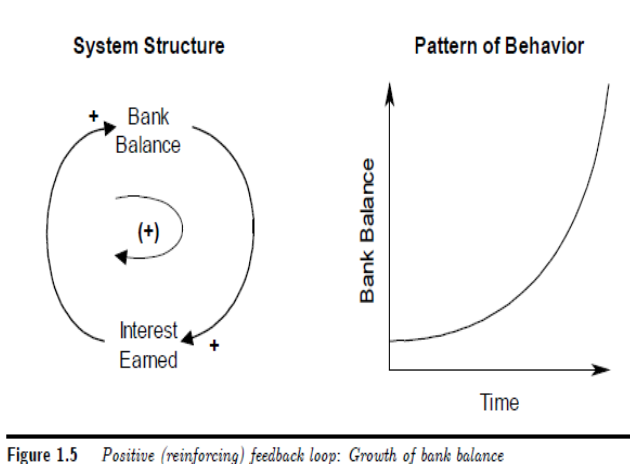
WHERE ARE THE COWS? “A prestigious economics journal contained an article which presented a model that was designed to forecast milk production in the US. By all statistical measures of validity the model was quite sound ... $\text{Milk Production} = a \text{ GNA} + b \text{ Interest rates} + \dots$ The equation. ... assumes that the dependent variable Milk Production is a function of a set of macroeconomic variables ... does not purport to represent how milk is *actually* produced ... **for nowhere in the equation do we see any cows!**”

Systems dynamics holds that the **structure** of the system -- the nodes and arrows in flow diagram -- is more informative than correlations or regressions. If you know the structure and can get reasonable estimate of parameters you can predict better and identify places to intervene. Causal model not data-mongering. This works when:

1. The way an organization/operation is **put together** determines its behavior.

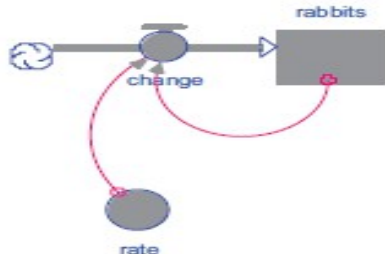
Example: The structure of AFL-CIO (leadership selected by union presidents) \rightarrow the way the organization behaves, so that changes in leadership – from Kirkland to Sweeney to Trumpke – produce only modest changes in union actions. Why? Union presidents concerned about their own fiefdoms can block almost anything. Same is true of lots of large bureaucracies – big firms, universities, etc.

2. When the behavior fits some **generic process**: oscillations, exponential growth, or decline that allow you to use a well-established model or pattern of feedback loops to describe data.



THREE GENERIC MODELS:

1) BIRTH-DEATH MODEL: $POP = POP(-1) + \text{Births} - \text{Deaths}$ is identity. Then make Births = $bPOP(-1)$, a positive feedback – INFLOW of units from STOCK where feedback is variable to itself. With just births, this leads to compound growth. $POP = (1+b) POP(-1)$ or $dPOP/POP(-1) = b$. percentage growth of population due to births is an



“Stella” has several levels. By clicking on a down arrow you can move to the equation level to see the mathematics and structure of the model. For the population growth model above the structure is:

```
rabbits(t) = rabbits(t-dt) + (change)*dt
INT rabbits = 100 (a specification I choose for this example)
INFLOWS:
  change = rate*rabbits
```

Deaths treated similarly: Deaths = $-dPOP(-1)$. This is a draining process. OUTFLOW of units from STOCK.

The birth-death process is a difference/differential equation: $POP - POP(-1) = (b-d) POP(-1)$, which is stable with an equilibrium when $b=d$, $\rightarrow \Delta \ln POP = 0$. Otherwise, it generates exponential growth or decline.

Highly unlikely to get $b=d$ if random draw from some distribution of numbers, so if we expect some stability in the size of the Population, make these parameters vary with other things in the model. For example, assume that birth rate declines as the population grows:

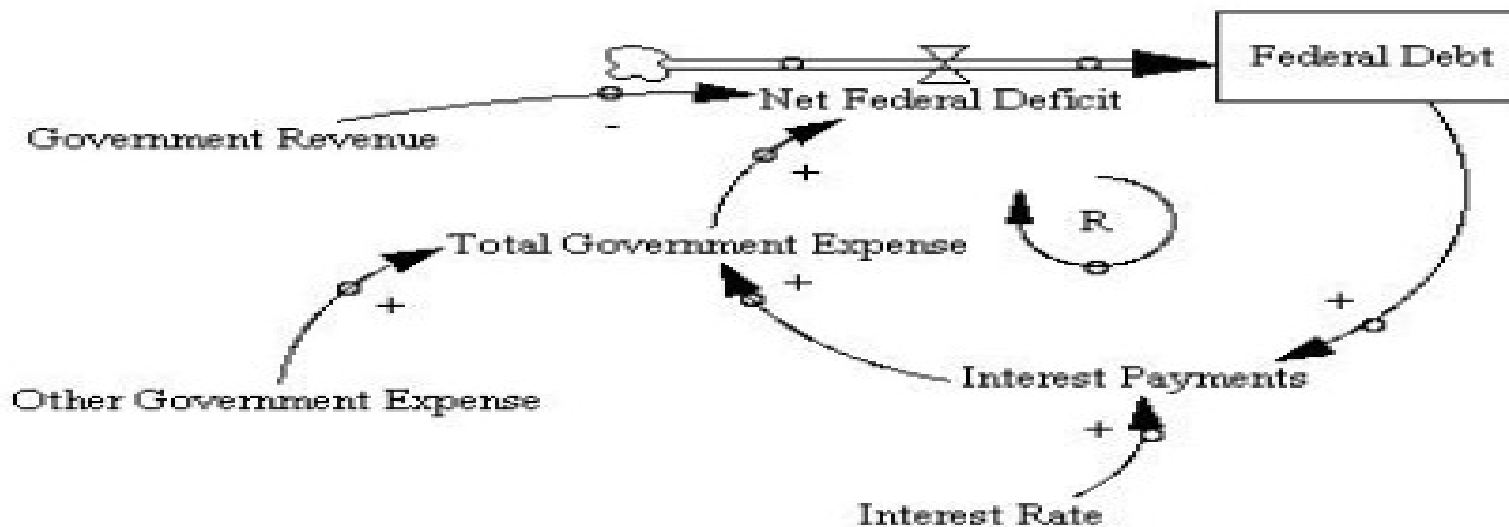
$b = c - c'POP(-1)$. This gives a NON-LINEAR/2nd order difference equation: $dPOP = (c-d) POP(-1) - c'POP(-1)^2$

Apply model to **Union growth/decline**. Birth is organization of a new work site, which occurs at rate g ; Death is closure of union plants, which occurs at rate r . Then we have Union membership U is: $U = (g-r) U(-1)$. To change this into a union density (UD) divide both sides by employment (E) so equation becomes $U/E = UD = (g-r)[U(-1)/E]$. Since $UD(-1) = U(-1)/E(-1)$, $U(-1)/E = UD(-1) E(-1)/E$.

If $E = (1+e) E(-1)$ where e is growth rate of employment, then $E(-1)/E = 1/(1+e)$ which for small $e \sim -e$ (if e is 1%, $1/(1+.01)=0.99$; if e is 3%, $1/1.03 = 0.97$; and so on.

So we have $UD = \sim (g-r-e) UD(-1)$. Density grows if $g-r>e$; is constant if $g-r \sim e$, and falls if $g-r<e$.

Government deficit and debt is similar. Debt is stock. Deficit is flow. To stabilize Deficit/GDP same as stabilizing U/E or POP . But the more meaningful ratio to stabilize is Debt/GDP so do some judicious divisions.

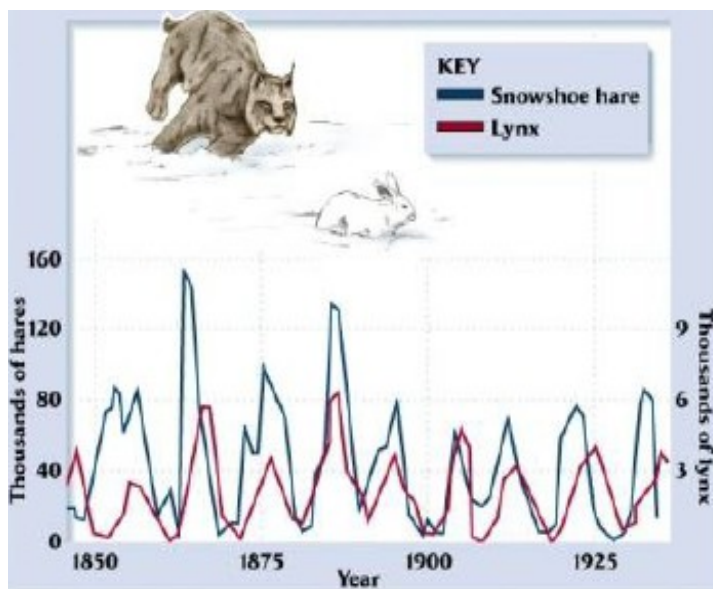
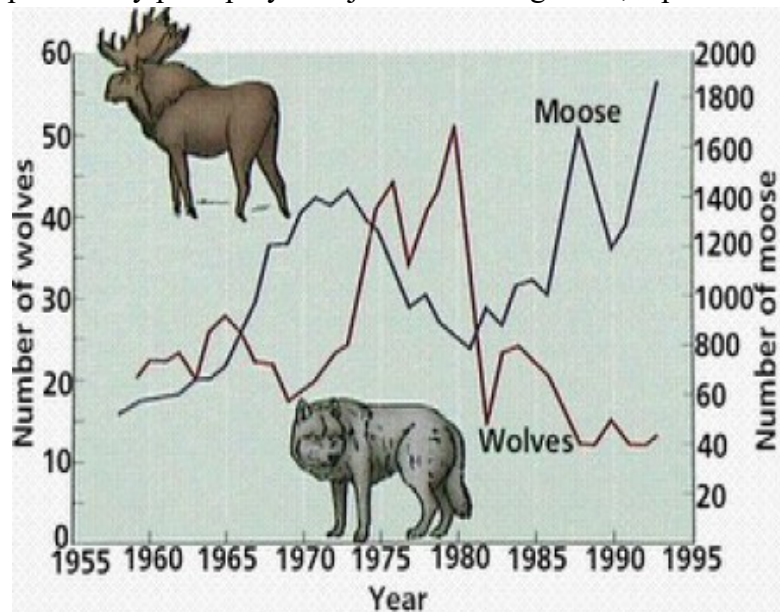


Model II: PREDATOR (y) PREY (x) model with two interacting stocks:

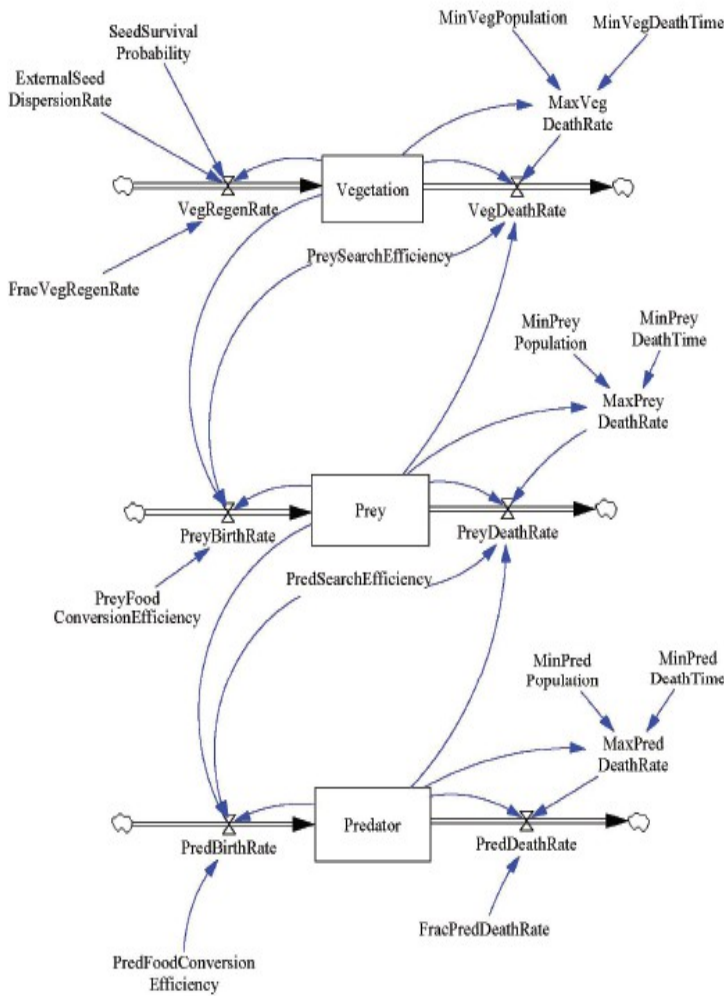
Prey: $dX/dt = (a-bY) X = aX - bXY$. Interpretation – a as birth rate and b as death rate from running into predator or in growth rate form: $dX/X = a-bY$ birth rate minus death from meeting predator

Predator: $dY/dt = -(\lambda-uX)Y = -\lambda Y + uXY$. Interpretation λ is death rate and birth/growth comes from eating up prey or in growth rate form: $dY/Y = -\lambda + uX$. Die unless you meet prey

Model can be represented by a cellular automata. Sites can be predator, prey, or empty. If a predator is adjacent to prey, the prey becomes predator with probability r or empty with $1-r$. If no prey adjacent to predator, predator dies with probability p . If prey is adjacent to bare ground, reproduces with probability q . Anything adjacent to bare moves there.



A Three stock version from Vensim (<http://www.cleanmetrics.com/pages/WildlifeManagement-2.pdf>)

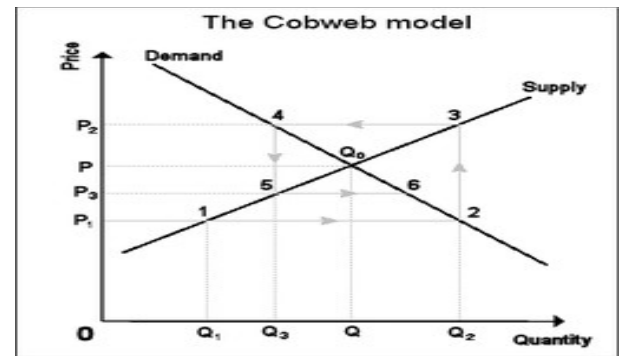
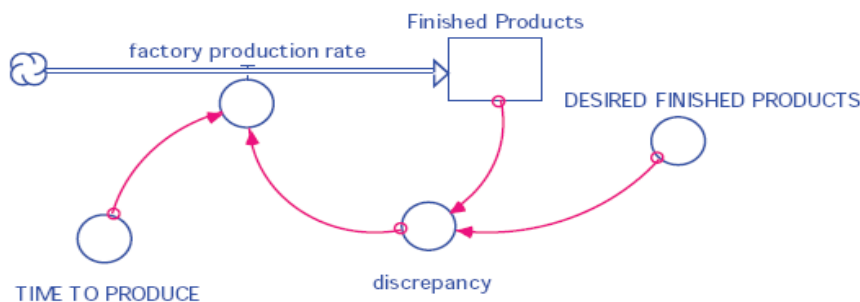


Vensim Model Equations

- (15) $\text{Predator} = \text{INTEG} (\text{PredBirthRate} - \text{PredDeathRate}, 10)$ Units: pred
 (16) $\text{PredBirthRate} = \text{PredFoodConversionEfficiency} * \text{PredSearchEfficiency} * \text{Prey} * \text{Predator}$ Units: pred/Month
 (17) $\text{PredDeathRate} = \text{MIN}(\text{FracPredDeathRate} * \text{Predator}, \text{MaxPredDeathRate})$ Units: pred/Month
 (18) $\text{PredFoodConversionEfficiency} = 0.1$ Units: pred/prey
 (19) $\text{PredSearchEfficiency} = 0.02$ Units: 1/pred/Month
 (20) $\text{Prey} = \text{INTEG} (\text{PreyBirthRate} - \text{PreyDeathRate}, 500)$ Units: prey
 (21) $\text{PreyBirthRate} = \text{PreyFoodConversionEfficiency} * \text{PreySearchEfficiency} * \text{Vegetation} * \text{Prey}$ Units: prey/Month
 (22) $\text{PreyDeathRate} = \text{MIN}(\text{PredSearchEfficiency} * \text{Predator} * \text{Prey}, \text{MaxPreyDeathRate})$ Units: prey/Month
 (23) $\text{PreyFoodConversionEfficiency} = 0.005$ Units: prey/veg
 (24) $\text{PreySearchEfficiency} = 0.2$ Units: 1/prey/Month
 (25) $\text{SAVEPER} = \text{TIME STEP}$ Units: Month [0,?]
 (26) $\text{SeedSurvivalProbability} = 0.25$ Units: Dmnl
 (27) $\text{TIME STEP} = 1$ Units: Month [0,?]
 (28) $\text{VegDeathRate} = \text{MIN}(\text{PreySearchEfficiency} * \text{Prey} * \text{Vegetation}, \text{MaxVegDeathRate})$ Units: veg/Month

Generic Model of ADJUSTMENT: link between DESIRED and ACTUAL gives logistic approach to equilibrium:

Assume you are initially short of DESIRED VALUE so that discrepancy is positive. Then more production reduces discrepancy → Minus sign to that loop. Discrepancy falls, then lower production rate → positive sign to that loop. The process is one of adjusting to make discrepancy zero or close to zero. The total adjustment loop is a negative feedback



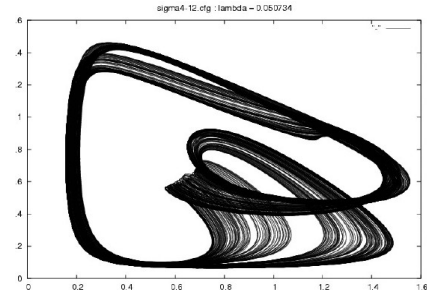
General Model IV: Cobweb. Consider the supply of college graduates, CG, which depends on the salary of graduates relative to less educated person in the preceding year $S(-1)$: $CG = A + a S(-1)$, where A is a shift term. Demand for graduates is $CGd = D - b S$, where D is shift term. This market equilibrates by supply = demand. So $D - b S = A + a S(-1)$. Thus, $S = D - A - a/b S(-1)$, which oscillates depending on the values of a/b. The assumption that CG depends solely on $S(-1)$ can be modified in many ways to allow for more sophisticated forecasts of the future.

Empirical work on Predator/Prey model.

Consumption of a single predator critical since it determines both the prey death rate and the predator rate of increase (usually modeled as proportional to the rate of feeding). But consumption rate of a single predator \rightarrow competition among predators that is missing from the Lotka-Volterra predator equation of $s \frac{dY}{Y} = -\lambda + uX$, where predator consumes prey as linear relation of # prey. But if there are lots of predators they will compete for prey. So how about modifying Lotka-Volterra by making birth of predator be uX/Y instead of uX ? This says should analyze in relative measures rather than absolute (Arditi & Ginsberg, How Species Interact).

PREY equation: $dX/dt = aX - bXY$ (as before) but PREDATOR $dY/dt = -\lambda Y + uX/Y$

In L-V predator growth depends only X the prey \rightarrow prey-dependent. A&G growth depends on X/Y -- ratio-dependent



Building a Systems Dynamics Model

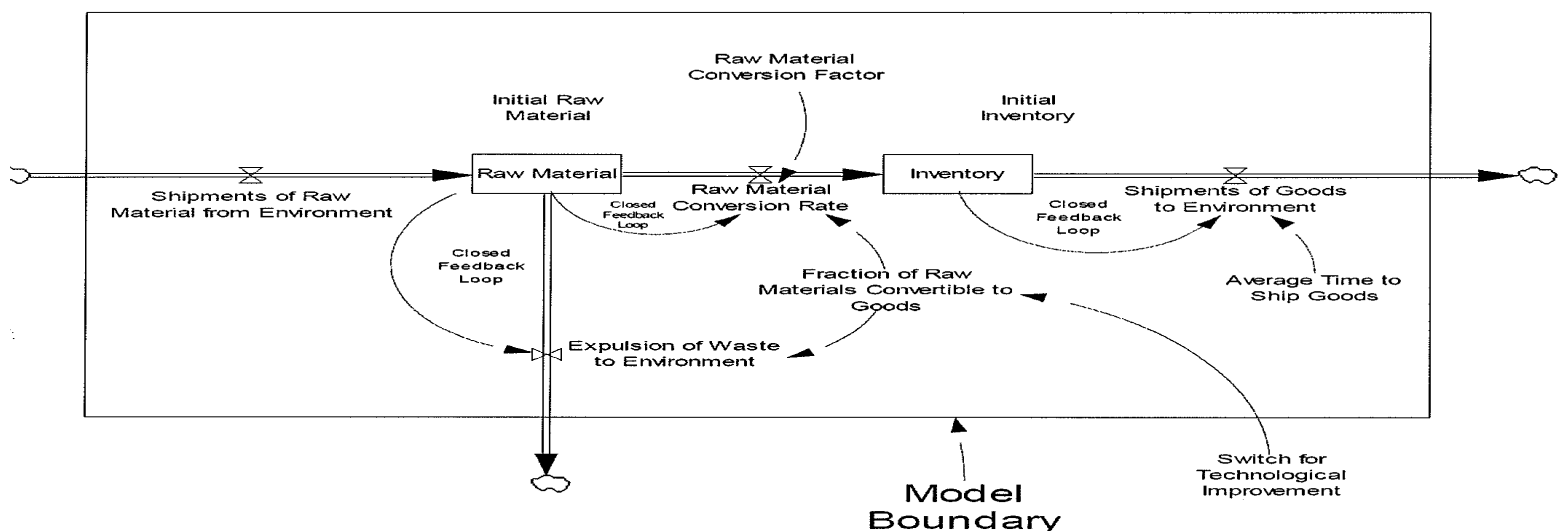
Systems dynamics uses arrows to represent difference equations. Feedbacks make an outcome depend not simply on the immediate equation but on the entire system. The systems dynamics program imposes internal consistency and a set of GENERIC PROCESSES that you can call on to yield a set of potential outcomes:

exponential growth (positive or negative feedbacks) ; stable equilibrium (sufficient negative feedbacks)
multiple equilibrium (some stable, some not); chaos (all you need is simple logistic)
oscillations – cobwebs or overshoot and collapse

A **STOCK** is a BOX/Reservoir that absent flows lasts from period to period. It can be qualitative -- esteem; thinking capacity. It can be a price. Stocks are slow-moving. Can decline with use -- inventory; consumables -- or with time --depreciation . Other stocks do not change or maybe grow -- stock of useful knowledge.

A **FLOW** is an arrow in a stock- flow diagram – with a circle/valve that is attached that leads to box that represents a stock. It reflects a differential/difference equation that changes a stock. The flow can be in both directions -- biflow. The flow is influenced by stocks -- the change in population depends on the population and the birth rate.

Flows can allocate a fixed stock among various categories. In a model of a union/firm, there will be some resources/ cash that will be allocated among different departments/functions, including for the firm dividends. A flow begins with a cloud or can end in a cloud, meaning where it comes from or where it goes is outside the model. Here is a **production function represented by a flow (compare to $Y=F(K,L)$)**:



Changing stocks through flows directs attention at the process, the key parameters, and opens the door for more subtle relations, for instance, flow depending on multiple factors. Flows turn static picture into dynamics. The final element are **converters/auxiliary variables**. These are arrows that convert things, such as units, and make it easier to specify and later modify parameters. These will often take the form of ratios.



Figure 1.1: Causal links and feedback loops: a positive causal link (left hand side), a negative feedback loop (middle), and a positive feedback loop (right hand side)

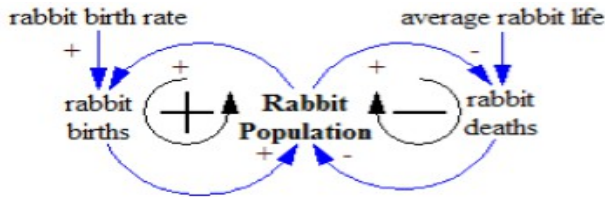


Figure 1.2: Example of a Causal Loop Diagram

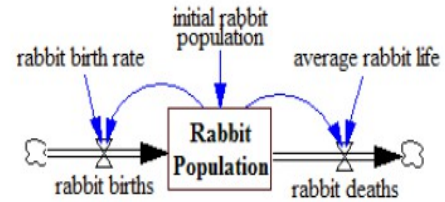
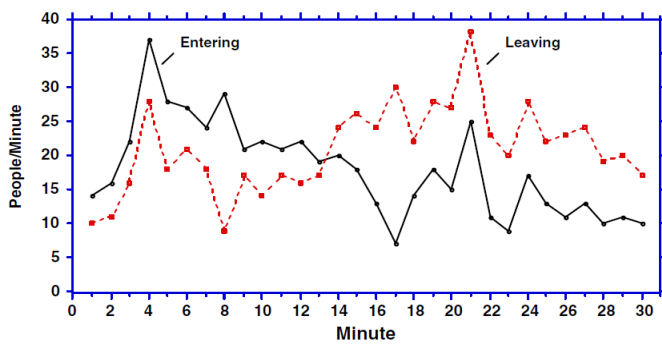


Figure 1.3: Example of a Stock-Flow Diagram

Systems dynamics models are based on “causal links” <http://stella-trial.software.informer.com/10.0>

THE PROBLEM OF SF FAILURE:

The graph below shows the number of people *entering* and *leaving* a department store over a 30-minute period.



Please answer the following questions.

Check the box if the answer cannot be determined from the information provided.

1. During which minute did the most people enter the store?

Minute _____ ☐ Can't be determined

2. During which minute did the most people leave the store?

Minute _____ ☐ Can't be determined

3. During which minute were the most people in the store?

Minute _____ ☐ Can't be determined

4. During which minute were the fewest people in the store?

Minute _____ ☐ Can't be determined

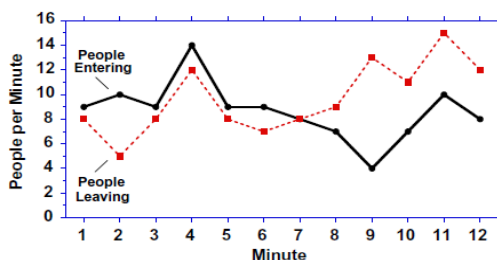
A total of 173 students enrolled in a graduate course in systems thinking and simulation at the MIT Sloan School of Management were given the department store task in Fig. 1. Participants were primarily MBA students and graduate students from other MIT departments or from Harvard University. The mean age was 29 (range 21–46) and 78% were male. All had taken calculus, and most had strong mathematics training: 71% had a degree in science, technology, engineering, or mathematics (STEM); 28% had a degree in the social sciences, primarily economics. Fully 40% had a prior graduate degree, most in technical fields. Students did the task in

Table 1

Results of the baseline department store task

	Most entering		Most leaving		Most in store		Fewest in store	
	N	%	N	%	N	%	N	%
Max entering t = 4	166	96.0	0	0	6	3.5	1	0.6
Max leaving t = 21	2	1.2	164	94.8	1	0.6	3	1.7
Max in store t = 13	0	0	0	0	76	43.9	4	2.3
Fewest in store t = 30	0	0	0	0	1	0.6	54	31.2
Max net inflow t = 8	4	2.3	0	0	50	28.9	0	0
Max Net Outflow t = 17	0	0	6	3.5	6	3.5	51	29.5
Initial in store t = 1	0	0	0	0	0	0	12	6.9
Cannot be determined	0	0	0	0	29	16.8	43	24.9
Other	1	0.6	2	1.2	2	1.2	2	1.2
No answer	0	0	1	0.6	2	1.2	3	1.7

A. Line graph



C. Table

Minute	People Entering	People Leaving
1	9	8
2	10	5
3	9	8
4	14	12
5	9	8
6	9	7
7	8	8
8	7	9
9	4	13
10	7	11
11	10	15
12	8	12

Question 1: Most entering?
Question 2: Most leaving?
Question 3: Most in store?
Question 4: Fewest in store?

Overall success rate (N = 264)
89%
83%
56%
46%

Why don't well-educated understand accumulation? Cronin, Gonzalez and Sterman Organizational Behavior and Human Decision Processes 108 (2009) 116–130 They say SF failure appears to be rooted in failure to appreciate the most basic principles of accumulation, leading to the use of inappropriate heuristics where many erroneously assuming that the behavior of a stock matches the pattern of its flows. Fischer and Gonzalez Making Sense of Dynamic Systems: How Our Understanding of Stocks and Flows Depends on a Global Perspective , Cognitive Science 40 (2016) 496–512) reduce SF failure by (a) a global as opposed to local task format; (b) individual global as opposed to local processing styles; and (c) global as opposed to local perceptual priming. These results say local processing explains SF failure.

Lots of programs to help you do systems dynamics modeling (Wikipedia)

Package name	Licensing	Implementation language	Last update (year)	More info
AMESim	Proprietary, commercial	C	2017	Distributed by Siemens. Modelica-models supported.
Analytica	Proprietary, commercial, free limited version	C++	2018	Supports system dynamics, Monte Carlo simulation for uncertainty, array abstraction for handling multidimensional data, linear and non-linear optimization. Uses influence diagrams to define, navigate, and document models.
AnyLogic	Proprietary, commercial, free Personal Learning Edition (PLE) for education, formal or not	Java	2017	Supports system dynamics, agent based and discrete event modeling, allows making hybrid models.
ASCEND	Free, GNU General Public License (GPL)	C	2012	For solving small to very large mathematical models, systems of non-linear equations, linear and nonlinear optimisation problems, dynamic systems expressed as differential-algebraic equations.
Berkeley Madonna	Proprietary, shareware	C++, Java	2016	Developed on Berkeley campus under sponsorship of NSF and NIH, used by academic and commercial institutions to build mathematical models for research and teaching. Solves ordinary differential equations (initial conditions and boundary value problems), difference equations (initial conditions and boundary value problems), multi-dimensional transcendental algebraic equation roots, discrete simulations using conveyors, ovens, and queues. Suitable for large-scale systems, boundary value problems, Monte Carlo models, curve fitting, root finding, batch processes, parameter plots, stiff systems, etc.
DYNAMO	Proprietary, no longer distributed commercially	AED, Pascal	1986	Historic DYNAMO models are often available at the MIT system dynamics website. DYNAMO software for microcomputers may be available via eBay or other resale sites.
GoldSim	Proprietary, commercial	C++	2016	Differs from traditional system dynamics approaches in that 1) it puts much greater emphasis on probabilistic simulation techniques to support representation of uncertain and/or stochastic systems; and 2) it provides a wide variety of specialized model objects (beyond stocks, flows and converters) in order to make models less abstract (and hence more transparent) and help represent processes and events that cannot easily be represented using a traditional system dynamics approach. These differences are due to the fact that GoldSim is primarily used for engineering and scientific applications where quantitative probabilistic predictions of future performance are required.

Package name	Licensing	Implementation language	Last update (year)	More info
iThink	Proprietary, commercial	C++, Javascript	2018	System dynamics and discrete event modeling with some agent-based capabilities. Drag and drop user interface builder allows simulations to be published online. Includes multilevel hierarchical models, reusable modules, multidimensional arrays, optimization, and Monte Carlo analysis.
MapleSim	Proprietary, commercial	Java (GUI), C, Maple (engine)	2017	Modelica-based system-level modeling tool. Leverages symbolic computing via tight integration with Maple.
NetLogo	Free, GPLv2	Java, Scala	2018	Agent-based modeling environment in LOGO; supports system dynamics models as a secondary feature.
OpenModelica	Free, QSMC Public License, Eclipse Public License (EPL), GNU General Public License (GPL)	C, C++, MetaModelica	2017	Modelica-based modeling and simulation environment. Modelica is an object-oriented, declarative, multi-domain modeling language for component-oriented modeling of complex systems; there is a freely available System Dynamics library (https://www.modelica.org/libraries).
Powersim Studio	Proprietary, commercial, free limited version	C++	2018	Supports system dynamics; building graphical diagrams using stocks and flow, including delays and feedback for non-linear models. Supports units, multi-dimensions running scenario simulations and Monte Carlo simulations.
Simcad Pro	Proprietary, commercial, free full featured demo	Visual interface	2019	Interactive process simulation software with support for manufacturing, healthcare and supply chain. 2D and 3D visualization with VR capability demo
Simantics System Dynamics	Free, Eclipse Public License (EPL)	Java, Modelica	2018	Free and open source system dynamics modelling software with stock and flow modelling, hierarchical models and array variables.
Simulink	Proprietary, commercial	?	2015	Tight integration with MATLAB
Stella	Proprietary, commercial, limited free online version	C++, Javascript	2018	System dynamics and discrete event modeling with some agent-based capabilities. Drag and drop user interface builder in Architect versions allows simulations to be published online. Includes multilevel hierarchical models, reusable modules, multidimensional arrays, optimization, and Monte Carlo analysis.