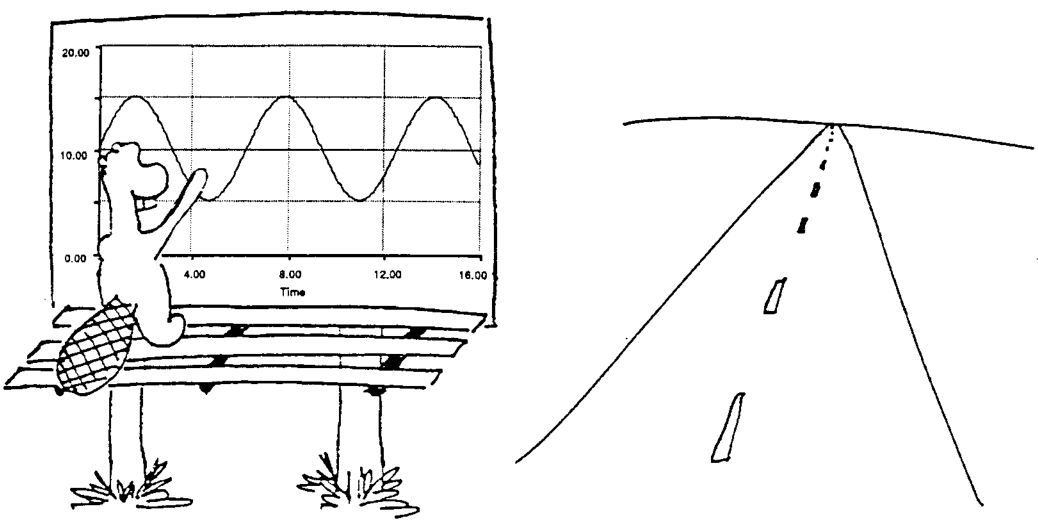
Road Maps 9

A Guide to Learning System Dynamics



System Dynamics in Education Project

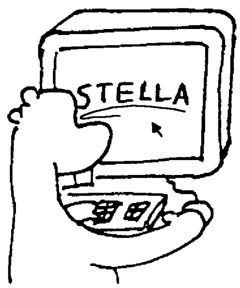
Road Maps 9

System Dynamics in Education Project
System Dynamics Group
Sloan School of Management
Massachusetts Institute of Technology

November 10, 1997
Latest Revision March 18, 2001

Compiled under the direction of Professor Jay W. Forrester

Welcome to Road Maps Nine!



Road Maps is a self-study guide to learning the principles and practice of system dynamics. This chapter is the ninth in the Road Maps series. Road Maps One through Three give a broad introduction to the field of system dynamics, focusing on understanding the structure and behavior of systems through positive and negative feedback loops. Road Maps Four through Six introduce several generic structures, give a taste of policy analysis and model validity, and provide opportunities for independent modeling. Road Maps Seven and Eight explain some common mistakes and misunderstandings in system dynamics modeling, offer exercises for improving modeling skills, analyze higher-order positive feedback loops as well as oscillating systems, and introduce sensitivity analysis.

Road Maps Nine improves your understanding of simple economic systems by modeling the use of credit cards. Another mistakes and misunderstandings paper illustrates how to correctly formulate robust table functions. Road Maps Nine then presents an important generic structure producing the behavior known as "overshoot and collapse." The chapter also explains how to perform graphical integration qualitatively, without any calculations. The chapter concludes with a paper by Professor John Sterman about using various types of computer models.

Topics Covered in Road Maps Nine

Improving Modeling Skills

- *The Credit Card Model* (D-4683)
  by Manas Ratha

- *Mistakes and Misunderstandings: Table Functions* (D-4653)
  by Leslie A. Martin

Generic Structures

- *Generic Structures: Overshoot and Collapse* (D-4480)

by Lucia Breierova

Graphical Integration Exercises

- *Graphical Integration Exercises Part 5: Qualitative Graphical Integration* (D-4675)

by Manas Ratha and Helen Zhu

Computer Models

- *A Skeptic's Guide to Computer Models* (D-4101-1)

by John D. Sterman

Things You'll Need for Road Maps Nine

Modeling Software

In order to complete Road Maps Nine and subsequent Road Maps, you will need to have access to modeling software. The Road Maps guides and most papers included in Road Maps were written with the use of STELLA II for the Macintosh. STELLA II is currently available for both the Macintosh and the Windows platforms. If you have any questions about STELLA, contact High Performance Systems (see Appendix). Ask about prices for educational use.

Vensim, Powersim, and DYNAMO are other software programs designed for building system dynamics models. Vensim is produced by Ventana Systems, which offers a free introductory version of its software, Vensim PLE, that can be downloaded off the World Wide Web. See the Appendix for more information about obtaining Vensim and Powersim.

Notice written June, 2000:

We have written a guide on how to use Vensim modeling software for each section of the Road Maps series that involves computer modeling. Each guide is located in the back of the exercise document. When Chapters 1-9 of the Road Maps series were written, STELLA software was the most common beginner modeling program available. Now you may choose from a number of system dynamics modeling software packages. If you would like more information on

Vensim, please go to http://www.vensim.com. A free version called Vensim PLE is located there.

For more detailed information on using Vensim software in the Road Maps series, please refer to the paper titled: "Vensim Guide (D-4856)" in the Appendix section at the end of Road Maps.

From now on as additional papers for the Road Maps series are written, the Vensim software will be used exclusively for modeling exercises.

A Computer

To run the latest version of STELLA, STELLA 5.0, on a Macintosh, you will need an Apple Macintosh computer (68020 processor or higher) with at least 8 MB of RAM, a 12 MB hard disk and System 7.1 or higher. To run STELLA 5.0 for Windows you will need an IBM PC-compatible computer with a 486-class processor running Windows 3.1 or greater. You will need at least 8 MB RAM, a hard disk with a least 16 MB of free space. Previous versions of STELLA have similar requirements.

In either case, if you plan on continuing to model, it may be a good idea to have access to a computer with more memory, hard disk space and a faster processor.

How to Use Road Maps Nine

Road Maps Nine explores several topics in system dynamics through selected readings and exercises. Before each reading or exercise is a short description of the reading and its most important ideas. After each reading or exercise, we highlight the main ideas before moving on.

Each chapter in Road Maps contains readings that introduce and strengthen some of the basic concepts of system dynamics. Other readings focus on practicing the acquired skills through various exercises or simulation games. Many of the chapters conclude with a prominent paper from the literature in the field of system dynamics.

As part of the spiral learning approach used in Road Maps, many concepts will be briefly introduced early on and then explained later in greater detail. Road Maps contains several series of papers that are spread out over successive chapters. Each of these series focuses on a specific topic or skill in system dynamics. The series start with a simple paper, and progress to further develop the idea in subsequent chapters.

Road Maps present the fundamental concepts of system dynamics as *System Principles*. These principles are enclosed in boxes that highlight them from the rest of the text to emphasize their importance. The progression of system principles in Road Maps allows you to revisit each principle several times. Each time a principle is revisited in Road Maps, you will build upon your previous understanding of the principle by learning something new about it. The system principles are the core of Road Maps around which the readings, exercises, and papers are built.

Now let's get started!

Improving Modeling Skills

More and more often, people use their credit cards when making purchases. Using a credit card is convenient because the user can spend money without having to carry cash, or even without having the amount of money necessary to make the purchase in his checking account. Unfortunately, the money borrowed on a credit card must be repaid, and, if it is not repaid on time, the credit card company that issued the card charges interest at a high rate. The following paper builds a simple model of using a credit card and examines the consequences of borrowing too much money on the borrower's quality of life.

- The Credit Card Model[1]

by Manas Ratha

This paper describes a simple system of purchasing goods with a credit card. It then leads the reader through the steps of building a system dynamics model: conceptualization, formulation of the model equations, editing the model, and evaluating the behavior of the model. In addition, the paper also explains the economic implications of using a credit card on the quality of life of the card user. The reader should build and simulate the model along with the paper.

[1] Manas Ratha, 1997. *The Credit Card Model* (D-4683), System Dynamics in Education Project, System Dynamics Group, Sloan School of Management, Massachusetts Institute of Technology, June 17, 29 pp.

Please read *The Credit Card Model* now and do the suggested exercises on your computer.

After reading *The Credit Card Model*...

The paper you just read provides an example of building a system dynamics model from scratch, starting from a system description. The paper leads the reader through conceptualization of the system, formulation and editing of the model equations, and finally, through simulation and evaluation of the model. Apart from improving modeling skills, the paper also shows some possible effects of overusing a credit card. Even though in the short run the quality of life of the card user improves, an unpaid balance is likely to decrease the user's quality of life in the long run. Thus, the system demonstrates a behavior characteristic of many complex systems: "short-term benefit, long-term cost."

The previous paper also reviews an important system principle that has already been introduced in Road Maps.

<u>System Principle #20:</u>

Rates are not instantaneously measurable.

No rate can, in principle, control another rate without an intervening level variable (as rates cannot be measured instantaneously and must be averaged by using a level for integration). Two rates may be directly connected in some models as a short cut, but the occasion is rare and should be avoided by inexperienced modelers.

Look back at section 5.3 of *The Credit Card Model*. For simplicity, the flow of "payments" is first set equal to the flow of "interest charges," as shown in Figure 4. Because it is impossible to measure the flow of "interest charges" instantaneously, the modeler instead introduces a variable called "interest on balance." Both flows, "payments" and "interest charges," are then set equal to "interest on balance." Good modeling practice does not set a flow equal to another flow.

We will come back to this system principle again in later chapters of Road Maps.

- Mistakes and Misunderstandings: Table Functions [2]

by Leslie A. Martin

This is the third in a series of papers in Road Maps that examine common mistakes and misunderstandings in system dynamics modeling. Using a simple population model, this paper explains that table functions should have dimensionless inputs and outputs and shows how to correctly formulate robust table functions.

Please read *Mistakes and Misunderstandings* now.
After reading *Mistakes and Misunderstandings...*

Table functions are an excellent tool for representing nonlinear relationships between model variables. For a table function to be dimensionally consistent, its input as well as its output should be nondimensional. In addition, table functions should be robust: they should cover the entire ranges of reasonable parameter values. As the next reading shows, many models of complex systems rely heavily on a correct formulation of table functions.

Generic Structures

Previous chapters of Road Maps presented several generic structures that are known for the behavior they produce. Such generic structures include S-shaped growth and oscillation structures. Many systems that generate another common type of behavior, overshoot and collapse, share the same basic structure, which is presented in the next paper.

- Generic Structures: Overshoot and Collapse [3]

by Lucia Breierova

This paper begins by modeling the structure and showing the behavior of two systems, a system of wells drilling petroleum on a field and a deer population system. Both models produce the type of behavior known as overshoot and collapse. Then, the generic structure that generates overshoot and collapse is

---

[2] Leslie A. Martin, 1997. *Mistakes and Misunderstandings: Table Functions* (D-4653), System Dynamics in Education Project, System Dynamics Group, Sloan School of Management, Massachusetts Institute of Technology, July 15, 13 pp.
[3] Lucia Breierova, 1997. *Generic Structures: Overshoot and Collapse* (D-4480), System Dynamics in Education Project, System Dynamics Group, Sloan School of Management, Massachusetts Institute of Technology, July 21, 40 pp.

presented, with special attention paid to the nonlinear table functions contained in the structure. The paper then offers two exercises in which the reader is asked to use the generic structure to model two different systems.

Please read *Generic Structures: Overshoot and Collapse* now and do the suggested computer exercises.

After reading *Generic Structures*...

Systems producing the behavior of overshoot and collapse contain a second-order negative feedback loop linking a nonrenewable resource to the growth of another stock. As the stock of interest grows, it consumes more and more of the resource that is incapable of regeneration. The decline of the amount of the resource affects the stock's growth, slowing down the growth until the stock is no longer able to grow because of the depletion of the resource. The stock starts declining and eventually collapses.

The paper you just read reviews an important system principles that you have already seen several times in Road Maps.

<u>System Principle #1:</u>

The feedback loop is the basic structural element of systems.

The more complex systems are assemblies of interacting feedback loops.

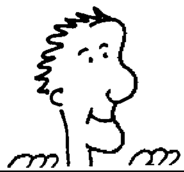
Consider the model of the Mayan population, shown in Figure 14 of *Generic Structures: Overshoot and Collapse*. In addition to the simple positive loop of "Population" and "births" and the negative loop of "Population" and "deaths," the Mayan population system contains several other feedback loops.

The loop relating the "State Of Environment" and "environment regeneration" is a positive feedback loop: the higher the value of "State Of Environment," the lower the "regeneration time," so the higher the rate of "environment regeneration," which further increases the value of "State of Environment."

The system, however, also contains another negative feedback loop, relating the "density" of the "Population" to the rate of "deaths" through the "damage to environment": the higher the "Population," the higher the "density," so the more "damage to environment" occurs. More "damage to environment" decreases the "State Of Environment," which in turn causes the "death fraction" and thus the "death rate" to increase, resulting in a stabilizing decrease of the "Population."

The structure of the Mayan population system is thus composed of several feedback loops that interact with each other to produce the dynamic behavior of overshoot and collapse.

The models presented in the previous paper also demonstrate another system principle, which you will see reviewed several times in later chapters of Road Maps. Remember from System Principle #22, introduced in Road Maps 8, that information links connect levels to the control of rates. Through information links, values of level variables go to the rate equations, determining the rates of flow. In the petroleum extraction model in *Generic Structures: Overshoot and Collapse*, information links determine the rate of "closing wells" directly from the level of "Wells" and indirectly from the level of "Petroleum Reserves."



System Principle #24:

Auxiliary variables lie only in the information links.

An auxiliary variable, or converter, is a subdivision of a rate equation. It allows a model to be disaggregated into easier to understand equation statements.

Look back at Figure 2 of *Generic Structures: Overshoot and Collapse* showing the model of petroleum extraction. The flow of "closing wells" is defined as:

closing wells = Wells * closing fraction (1)

The "closing fraction" is an auxiliary variable, the product of the "NORMAL CLOSING FRACTION" and the "effect of extraction on closing fraction" multiplier. Therefore, without using the auxiliary variable, one could also write the rate equation as:

closing wells = Wells * NORMAL CLOSING FRACTION * effect of extraction on closing fraction (2)

Equation (2), however, is more complicated and less understandable than equation (1), which contains the auxiliary variable "closing fraction." Thus, auxiliary variables help in the formulation of easily understandable rate equations.

Graphical Integration Exercises

Previous chapters of Road Maps presented several papers in which graphical integration was explained as a mathematical process used to determine the behavior of a stock from the graph of the behavior of flows into or out of the stock. The next paper in the Graphical Integration Exercises series treats graphical integration in a more intuitive way and explains how to estimate stock behavior without knowing the precise values of flows and without doing any calculations.

- Graphical Integration Exercises Part 5: Qualitative Graphical Integration [4]

---

[4] Manas Ratha and Helen Zhu, 1997. *Graphical Integration Exercises Part 5: Qualitative Graphical Integration* (D-4675), System Dynamics in Education Project, System Dynamics Group, Sloan School of Management, Massachusetts Institute of Technology, September 19, 33 pp.

by Manas Ratha and Helen Zhu

This paper explains how to determine, in general, a stock's behavior by looking at the graphs of the inflows and outflows from the stock, without performing any calculations. The paper focuses on linear flows (flows that can be graphed as straight lines) and explores seven possible types of flows. For each type of flow, the associated stock behavior is presented and explained. Examples and exercises with solutions are also offered to give the reader an opportunity to practice the skills acquired from the paper.

Please read *Graphical Integration Exercises* now and do the suggested exercises.

After reading *Graphical Integration Exercises*...

The paper you just read classifies linear flows into seven categories and describes the associated stock behavior. The sign of the flow determines whether the stock remains constant (if the flow is zero), grows (if the flow is positive), or declines (if the flow is negative) over time. The magnitude of the flow determines whether the stock changes at a constant, increasing, or decreasing rate. Qualitative graphical integration is important because it helps a modeler to intuitively understand the relationship between the behavior of a stock and the flows into or out of the stock.

Graphical Integration Exercises Part Five reviews an important system principle that was introduced in previous chapters of Road Maps:

<u>System Principle #4:</u>

Levels are accumulations (integrations).

Levels change smoothly but not instantaneously—there are no discontinuities, no jumps.

Look back at Example 3: Graphical Integration of Combined Flows in *Graphical Integration Exercises Part Five: Qualitative Graphical Integration.* The behavior of the net flow in the example contains two sudden, step changes. The value of the stock affected by the flow, however, cannot change instantaneously. Instead, when the slope of the stock graph changes as a result of the sudden change in the net flow, the value of the stock changes gradually, as shown in Figure 14.

Computer Models

Building computer models is an important part of system dynamics. System dynamics models, however, are not the only type of computer models. Various modeling techniques are now being used in an attempt to understand and forecast the behavior of complex systems. The following paper by Professor John Sterman explains the need for using computer models and examines the advantages and limitations of several types of models.

- A Skeptic's Guide to Computer Models[5]

by John D. Sterman

This reading offers a detailed description of computer modeling and of the advantages and disadvantages of models. It takes a close look at some important modeling techniques (optimization, simulation, and econometrics), focusing on their assumptions and the possible implications of their application in foresight and policy making.

Please read *A Skeptic's Guide to Computer Models...* now.

---

[5]John D. Sterman, 1991. *A Skeptic's Guide to Computer Models* (D-4101-1), System Dynamics Group, Sloan School of Management, Massachusetts Institute of Technology.

After reading *A Skeptic's Guide to Computer Models...*

The paper you just read emphasizes the advantage of using computer models over the use of mental models. Computer models, however, are not perfect, and must always be subjected to careful analysis before their conclusions can be used in policy making.

Several system principles about models can be drawn from the analysis in *A Skeptic's Guide to Computer Models*. We will return to them in later chapters of Road Maps. The first one explains the concept of abstract models.

System Principle #25:

Mathematical simulation models belong to the broad class of abstract models.

A model is a substitute for an object or a system. Some models are physical, such as a toy airplane or an architectural scale model. We are familiar with these. Some models are abstract. These abstract models include mental images, literary descriptions, behavior rules for games, and legal codes.

Mathematical simulation models also belong to the broad class of abstract models. Because computer modeling has become so widespread in recent years, it is important to understand the assumptions and applications of various modeling techniques.

The second system principle drawn from *A Skeptic's Guide to Computer Models* discusses model validity.

System Principle #26:

Model validity is a relative matter.

The usefulness of a mathematical simulation model should be judged in comparison with the mental image or other abstract model that would be used instead. No model is a perfect representation of a real object. A model is successful if it opens the road to improving the accuracy with which we can represent reality.

Finishing off Road Maps Nine

Road Maps Nine led you through building a simple system dynamics model of credit card spending and showed you how to formulate dimensionless and robust table functions. Your newly-acquired knowledge about table functions was then used in exercises in which you modeled systems exhibiting the overshoot and collapse behavior mode, using a simple generic structure. You also practiced your graphical integration skills in a more intuitive, noncomputational way. Finally, you improved your understanding of several types computer models and of models in general.

In Road Maps Nine, you were introduced to three new system principles:

a) that auxiliary variables lie only in the information links;

b) that mathematical simulation models belong to the class of abstract models; and

c) that model validity is a relative matter.

Key Terms and Concepts:

dimensionless inputs and outputs
econometric models
magnitude of a flow
model validity
multipliers
nonrenewable resource
normal values
optimization models
overshoot and collapse
quantitative vs. qualitative graphical integration
reference points
robustness
short-term benefit, long-term cost
simulation models
table function
value of a flow