The Delft Method for System Dynamics

Auping, Willem L.; d'Hont , Floortje; Kubli, M.D.; Slinger, J; Steinmann, P.; van der Heijde, Floris; van Daalen, C.; Pruyt, Erik; Thissen, W.A.H.

DOI
10.59490/tb.97

Publication date
2024

Document Version
Final published version

Citation (APA)
Auping, W. L., d'Hont , F., Kubli, M. D., Slinger, J., Steinmann, P., van der Heijde, F., van Daalen, C., Pruyt, E., & Thissen, W. A. H. (2024). *The Delft Method for System Dynamics*. TU Delft OPEN Publishing. https://doi.org/10.59490/tb.97

The Delft Method for System Dynamics

Willem Auping, Floortje d'Hont, Merla Kubli, Jill Slinger, Patrick Steinmann, Floris van der Heijde, Els van Daalen, Erik Pruyt, Wil Thissen

TU Delft OPEN

The Delft Method for System Dynamics

W.L. Auping, F.M. d'Hont, M.D. Kubli, J.H. Slinger, P. Steinmann, F.T. van der Heijde, C. Van Daalen, E. Pruyt, W.A.H. Thissen

Version: November 2024

Colophon
The Delft Method for System Dynamics

Authors:
W.L. Auping[1], F.M. d'Hont[1], M. D. Kubli[1], J.H. Slinger[1], P. Steinmann[1], F.T. van der Heijde[1], C. Van Daalen[1], E. Pruyt[2], W.A.H. Thissen[1]
[1] Faculty of Technology, Policy and Management, Delft University of Technology
2 Center for Policy Exploration, Analysis and Simulation (PEAS)
Willem Auping w.l.auping@tudelft.nl 0000-0003-1898-643X
Floortje d'Hont f.m.dhont@tudelft.nl 0000-0002-6966-023X
Merla Kubli m.d.kubli@tudelft.nl 0009-0007-3948-3491
Patrick Steinmann p.steinmann@tudelft.nl 0000-0002-6204-6253
Jill Slinger j.h.slinger@tudelft.nl 0000-0001-5257-8857
Floris van der Heijde 0009-0003-6142-726X
Erik Pruyt 0000-0001-8332-5422
Els van Daalen 0000-0001-7635-572X
Wil Thissen 0000-0003-3288-4675

Contributions
W.L. Auping (Original Idea, Theory, Exercises, Review & Editing, Images), F.M. d'Hont (Original Idea, Theory, Exercises, Review & Editing, Images), M. D. Kubli (Theory, Review & Editing), J.H. Slinger (Review & Editing), P. Steinmann (Review & Editing), F.T. van der Heijde (Editing, Images), C. van Daalen (Theory, Images), E. Pruyt (Original Idea, Theory, Exercises, Images), W.A.H. Thissen (Theory).
The order of the authors is based on the legacy of Delft SD education materials and not on the relative size of their contributions.

Keywords:
system dynamics, education, simulation modelling, complex systems, vensim, stella, powersim

Published by:
TU Delft OPEN Publishing | Delft University of Technology, The Netherlands

DOI: https://doi.org/10.59490/tb.97

ISBN paperback/softback: 978-94-6366-893-4
ISBN eBook/PDF: 978-94-6366-894-1

Cover design made by Willem Auping
Copyright clearance made by the TU Delft Library copyright team

Preface

System Dynamics (SD) was developed in the 1960s by J. Forrester, since many of the existing methods for solving problems did not offer a sufficient understanding of strategic problems in complex systems. The objective of SD modelling is, therefore, to clarify the relation between the behaviour of a system as a function of time and the underlying processes, which is referred to as the system structure in SD. The method consists of observing and identifying the behaviour of a system as a function of time and developing a model with the possibility of explaining existing system behaviour and designing improved system behaviour (Wolstenholme, 1989a).

SD modelling can be used to understand complex systems in a wide variety of fields, such as healthcare, energy transition, environmental management, safety and security, public order, social dynamics, education, economics, business management, and operations and supply chain management. It is therefore essential for anyone working in these fields to have a basic understanding of what System Dynamics are, and gain some working skills in SD modelling.

This e-book introduces System Dynamics Modelling to novice modellers. It is specifically designed to support courses on simulation methods at the Faculty of Technology, Policy and Management (TPM) at Delft University of Technology (TU Delft) in the Netherlands, including *Systems Modelling 3* and *Introduction to TPM Modelling.* However, it can also be used by other institutions that follow a practice-oriented and skill-oriented approach to learning, while it also offers sufficient theoretical information to support autonomous learning.

We choose to publish the e-book under an open access license so that SD learners from all over the world can study the material, independent of their study programmes or university access.

Authors

The System Dynamics education team of TU Delft has developed and published this book. The material builds on decades of System Dynamic teaching experience at TU Delft, where Bachelor and Master students are trained in various methods for modelling and simulating complex challenges (Pruyt, 2013; Van Daalen, Thissen, & Phaff, 2006). All co-authors have contributed to the development of System Dynamics education at TU Delft. This e-book builds on earlier theory-oriented syllabi, materials drawn from Pruyt (2013) and experience built in decades of System Dynamics education.

The Delft Method: a teaching philosophy for learning by doing

Modelling is difficult and learning how to build simulation models is even harder. There are a many excellent books that provide methods for learning System

Dynamics (SD) (e.g., Bossel, 2007a, 2007b, 2007c; Coyle, 1996; Duggan, 2016; Ford, 2009; Meadows, 2008; Morecroft, 2015; Pruyt, 2013; Richardson & Pugh, 1981; Sterman, 2000; Warren, 2002). Most of these books primarily focus on understanding theory first and practicing modelling skills later (cf., Carver, 1996). Others only focus on theory and hardly offer any (modelling) exercises. This book offers a unique approach to learning SD modelling, and therefore is a valuable addition to other works already available.

The starting point of this is book is our conviction that modelling is best learnt by doing and practicing one's skills. This book, *The Delft Method for System Dynamics*, is inspired by the Dutch language learning method “Delftse Methode” (the "Delft Method", Sciarone & Montens, 1985), which is used in courses for internationals at TU Delft and beyond. In these language courses, students are encouraged to learn Dutch by doing, following what is often called a ‘natural’ approach to language learning. Instead of learning the rules first and make sentences later, students build their language skills by ‘doing’: by speaking, by writing, by making mistakes, by many repetitions and constant feedback. Theory, and grammar and spelling rules stand in service to active language abilities. The method is particularly well suited for kinaesthetic learners: students who prefer to learn based on experiences – moving, touching, doing (Fleming, 1995; Gardner, 1983). Owing to the engineering focus of TU Delft, students flourishing with this type of learning are particularly frequent. Students at TU Delft also often do not have much linguistic knowledge, making the Delft Method particularly useful.

This pedagogical philosophy inspired the System Dynamics modelling courses taught at TU Delft and this e-book. Indeed, there is overlap between learning spoken languages on the one hand and learning modelling and programming languages on the other. There are of course differences, not in the least the fact that spoken languages are vastly more complex, but the same language learning strategies can apply to both. In the Delft Method, the underlying logic of a language (the rules or ‘syntax’) is revealed through practice. In our translation of the Delft Method to System Dynamics, we apply the same method: learning by doing.

How to use this the e-book

Focus on modelling

In this e-book, we keep the theory sections short. Theoretical concepts are explained briefly and pragmatically; the emphasis, however, lies on the actual practice of modelling. Learners are encouraged to try the exercises early on, without fully understanding the theory. Exploring, struggling and reflecting are key components in the learning process. You are encouraged to try to do the exercises and don't worry if you struggle at bit and must find some things out by yourself.

Working together works

Modelling is not easy to do by yourself, although it is possible. We find that it works best if learners work together. It often helps to discuss things with classmates or in peer groups. You can also learn a lot from explaining things to others.

Try first, but help is not far away

If you get stuck, do not waste time worrying. Software documentation of the different modelling languages (e.g., Vensim, Stella, or Powersim) and fellow students can often be of help. For students following System Dynamics courses at TU Delft, teaching assistants (first line) and teachers (second line) are available for questions during the computer labs.

Follow best practices in modelling

It is important that you get used to good modelling practices. These are:

- Define all units as soon as you formulate a variable.
- Formulate the model variable by variable, equation by equation. Do not build the whole structure at once, before entering the equations. Finish the equation before you go to a next variable.
- Always have a running model.
- Save your model frequently. Backup versions may be helpful when modelling advances don't work out as planned.
- When you get stuck with modelling: take a walk.

Reading guide

Chapter 1 introduces the basic principles and background of System Dynamics.

Chapter 0 through Chapter 5 describe the exercises. Each exercise chapter consists of two parts: theory and modelling questions. The theory questions are qualitative exercises in which concepts like diagramming conventions, transferring stock-flow diagrams into CLDs, and recognising the order (1^{st}, 3^{rd}, or infinite) and type (material or information) of delays. You should be able to make the theory exercises without using SD software (e.g., Vensim, Stella), but it is not forbidden to use software to solve them. The theory exercises are in the form of short answer questions.

All modelling exercises contain a model. In this model description, all variables are given in *italics*. Emphasis lies on learners being able to distinguish different types of variables (e.g., stocks, flows and auxiliaries). The text generally only allows a single interpretation: if you model the description correctly, you should get a model that generates exactly the same behaviour as the exemplar. Also, the stock-flow structure of your model should be the same as the example model. For most exercises, answers are available (see Auping et al., 2024).

The theory in this book follows a typical modelling cycle: problem articulation and conceptualization are discussed in Chapter 0, model formulation in Chapter 0, evaluation including verification in Chapter 0, and model use and policy testing in Chapter 9.

We hope you will enjoy this book and share the joy of systems thinking and modelling with us. We prepared the material in this e-book with care and foresee regular updates to the material. Please don’t hesitate to contact us with any remarks or feedback.

Delft, August 2024,

The authors.

Contents

1. Introduction to System Dynamics

This chapter begins with a history and background of the System Dynamics (SD) field. It continues by explaining the basic principles of SD: stocks and flows, numerical methods, and feedback.

1.1 History and background

Many problems are too complex for people to understand. The lack of understanding can lead to 'curing' the symptoms rather than pin down the source of a problem, the so-called 'end of pipe solutions'. System Dynamics was developed by J. Forrester (1961) at the Massachusetts Institute of Technology (MIT), who felt that many of the existing methods for solving problems did not offer a sufficient understanding of strategic problems in complex systems. Forrester combined ideas from three, at the time relatively new, disciplines (Meadows, 1976):

- control engineering: the concepts of feedback and self-regulation,
- cybernetics: the nature of information and the role of information in control systems,
- organisational theory (management sciences): the structure of organisations and the decision-making processes.

From these basic ideas, Forrester developed a philosophy and a collection of representation techniques to model complex systems. The method focuses on questions regarding dynamic behaviour of systems. This was how System Dynamics was developed. The continuous and deterministic nature of System Dynamics distinguishes it from, for example, Discrete Event Simulation and Agent-Based Modelling, both stochastic simulation methods that work with events scheduling or ticks instead of continuous time steps.

The objective of System Dynamics modelling is to clarify the relation between the behaviour of a system as a function of time and the underlying processes, or system structure. The method consists of observing and identifying the behaviour of a system as a function of time and developing a model with the possibility of explaining existing system behaviour and designing improved system behaviour (Wolstenholme, 1989a).

During the first years of the development of the method, the applications were mostly industrial, for example in Industrial Dynamics (Forrester, 1961). Later, the applications became broader and included urban development (Urban Dynamics) (Forrester, 1969) and global development (World Dynamics) (Meadows, Meadows, Randers, & Behrens, 1972). Since then, the field has expanded significantly. The scope of individual studies has become smaller, but the area of application is broad. In addition, a shift occurred away from the obligatory use of quantitative models to an awareness of the relevance of the qualitative aspects of modelling (Wolstenholme, 1989b) .

Application domains in SD include, but are not limited to, health policy, energy transition and resources scarcity, environmental and ecological management, safety and security, public order and public policy, social and organisational dynamics, education and innovation, economics and finance, organisational and strategic business management, information science, and operations and supply chain management.

1.2 Basic principles

One of the basic principles of the System Dynamics philosophy is that the *behaviour* of a system is caused by the *structure* of the system. The structure not only contains all physical aspects of, for example, a production process, but also includes policies and traditions, both tangible and intangible, that are important to the decision-making process in the system (Roberts, 1978). Such a structure contains accumulation, delays and feedback. The *feedback* concept is an essential characteristic and major strength of SD. Considering feedback in systems prevents solutions based on linear reasoning, which often only highlight and treat the symptoms, but not the root cause.

Another aspect of the SD philosophy is the idea that organisations can best be considered in terms of underlying 'flows' and 'stocks'. For example, flows of people being born, migrating, or dying, increase or decrease the stocks of people.

1.2.1 Stocks and flows

SD models are in essence large sets of integral equations which are numerically solved, and can be depicted with SD-specific diagrammatic conventions. The most important elements in SD models are stocks, sometimes also called levels, which are connected by flows. The behaviour of a stock over time is mathematically defined as an integral equation:

$$S(t) = S(t_0) + \int_{t_0}^{t} \big(f(t) - g(t)\big) dt, \qquad \text{Eq. 1.1}$$

where $S(t)$ is a stock at time t, $S(t_0)$ the initial value of this stock, $f(t)$ an inflow and $g(t)$ an outflow. dt refers to delta time, the integration step. Besides stocks and flows, SD also knows auxiliary variables and constants.

Figure 1.1 shows a simple SD model, where constant $S_0 = S(t_0)$, flow $f(t)$ is a function of constant c_1 and $s(t)$, auxiliary $a(t)$ is a function of constant c_2 and $s(t)$, and flow $g(t)$ is a function of constant c_3 and auxiliary $a(t)$. As the interconnected set of integral equations can become too large to analytically solve, SD languages like Vensim (Ventana Systems, 2010) use numerical integration methods like Euler (Euler, 1768) and Runge-Kutta 4 (Kutta, 1901; Runge, 1895).

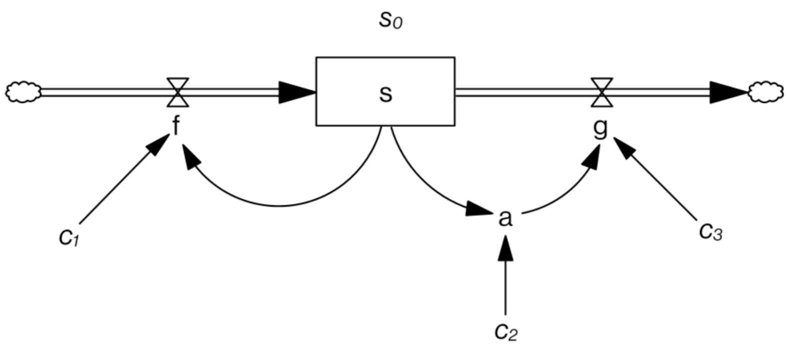


Figure 1.1. Simple stock-flow structure in SD diagrammatic conventions

1.2.2 Feedback

Feedback means that a chain of causal connections between variables exists. This influence can take place directly or indirectly, through other variables. Feedback for example exists in situations in which a decision-maker is influenced by the consequences of his or her actions. These consequences may become clear quickly or after some time; for example, if an increase in the number of cycling lanes results in the increased attractiveness of cycling. This in turn boosts the use of the cycling lanes, and, therefore, a greater need for new cycling lanes emerges. This may lead to a further expansion of the number of cycling lanes, closing the feedback loop. Closed loops and delays (for example the time period between orders and deliveries, between decisions and consequences of decisions, etc.) are characteristic for all feedback processes. In reality, people are often not aware of the fact that they play a part in a large number of different and complex social, economic and organisational feedbacks. The larger the delays in the feedback loop and the more indirect the consequences, the more difficulties people have in recognizing feedback structures (Roberts, 1978).

Feedback structures are briefly discussed below, using causal diagrams. A feedback *loop* consists of two or more connections between variables that are connected in such a way that if one follows the arrows starting at any variable in the loop, one eventually returns to the first variable. One loop will never contain the same variable twice. A feedback *system* consists of two or more connected feedback loops (Roberts, 1978). There are two types of feedback relations: positive or reinforcing, and negative or balancing feedback.

Reinforcing or positive or feedback

Reinforcing feedback loops result in a continuous increase or decrease of the value of related variables. Figure 1.2 shows two examples of reinforcing feedback loops. The curved arrows display a causal relationship between the variables, indicating the direction of the causality. A link marked “+” refers to a situation where, when one variable changes, the influenced variables change in the same direction, depicting a positive correlation/causality. A link marked "-" describes a negative correlation/causality, where the variables move in opposite directions.

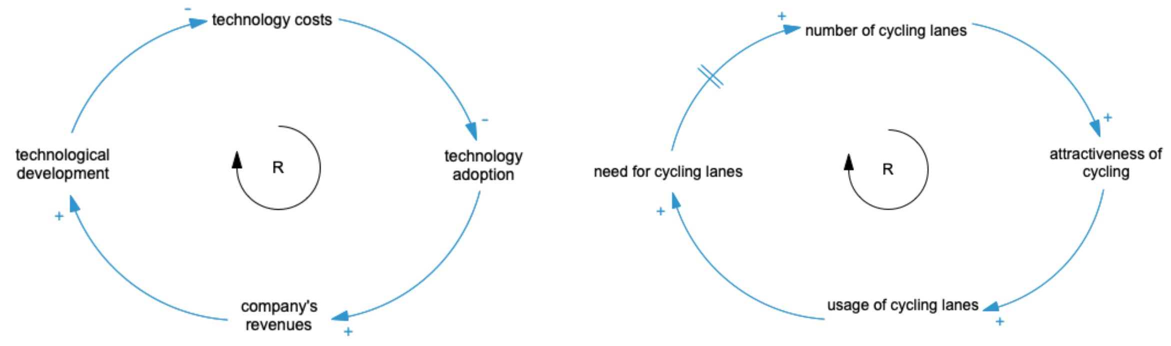


Figure 1.2. Two examples of reinforcing feedback loops

A feedback loop is called reinforcing (R) or positive (+), if an initial increase in variable A leads after some time to an additional increase in A and so on, or if an initial decrease in A leads to an additional decrease in A and so on. In isolation, such feedback loops are self-enhancing: they generate exponentially escalating behaviour which could be (extremely) beneficial or (extremely) detrimental. For this reason, some system dynamicists believe that the term reinforcing is preferable over the term positive, as it cannot be confused with its everyday meaning: vicious circles are in fact positive or reinforcing feedback loops, but highly undesirable ones (Wolstenholme, 1989b).

Balancing (or negative) feedback

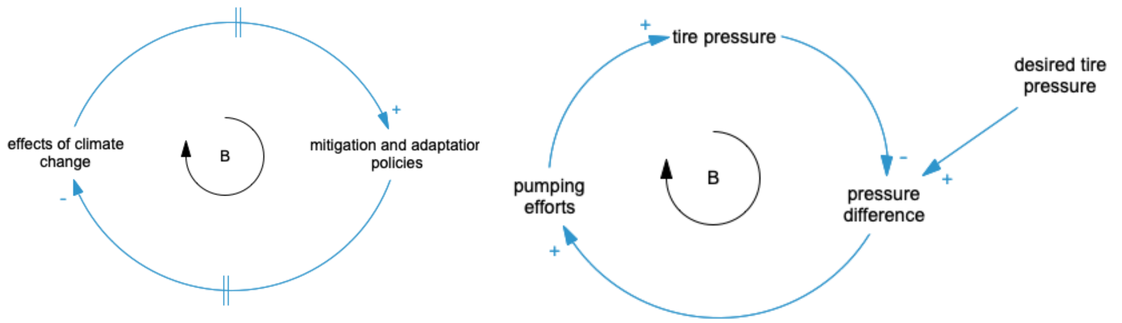


Figure 1.3. Two examples of negative feedback loops.

A feedback loop is called balancing (B) or negative (-), if an initial increase in variable A leads after some time to a decrease in A, or if an initial decrease in A leads to an increase in A. In isolation, such feedback loops generate balancing or goal-seeking behaviour. They are sources of stability as well as resistance to change. The presence of a balancing feedback loop in a system does not imply that the objective will be achieved nor that the process is under control. Balancing, or negative, feedback may also cause undesirable behaviour, for example undesirable oscillatory behaviour due to balancing feedback loops with delays.

Causal diagrams can be used to determine whether a particular loop is positive or negative. The net effect can be determined by multiplying the signs of all connections in the loop. If the net effect of the connections is negative, the entire loop is negative. Conversely, it can be stated that if the net effect is positive, the entire loop is positive.

An important notion in the feedback perspective of system dynamics is that no 'one' variable is solely responsible for all changes in the system. Variables that are part of a loop are both the cause and effect of change. In general, this means that problems generated by a system are not the responsibility of one single actor (Senge, 1990).

1.3 Modelling cycle

Model development and use is generally considered to be part of a "modelling cycle". These cycles emphasise the cyclical and iterative nature of model development, and distinguish multiple phases. In this book, we use the following steps in the modelling cycle: problem articulation, conceptualisation, formulation, evaluation, and policy testing.

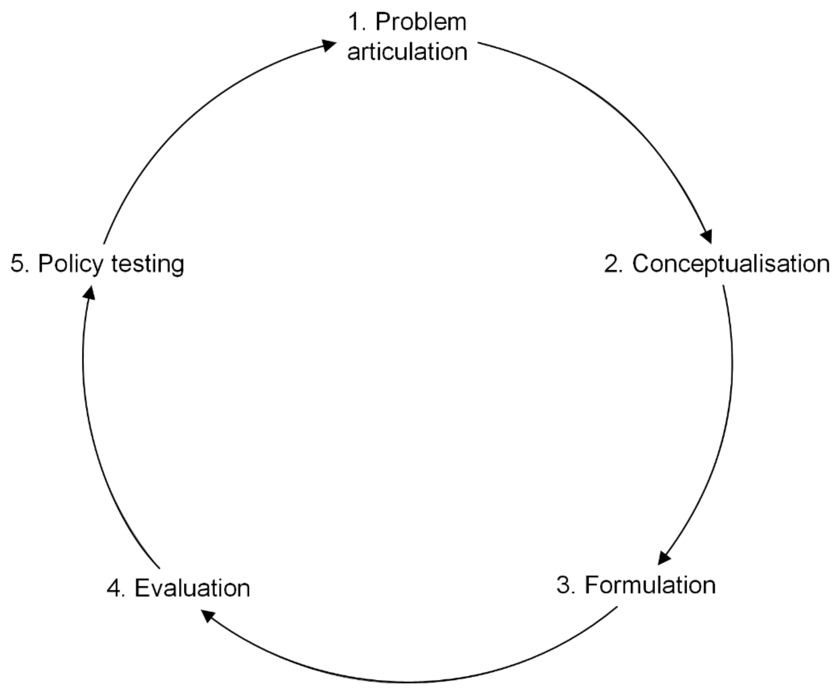


Figure 1.4. Modelling cycle

The **problem articulation** phase is generally considered to be the first phase of the model development cycle. The overall goal of this phase is to formulate a research design. This design can be obtained by taking the following steps: problem formulation, boundary or scope selection, time horizon selection, and, if desired, the identification of potential reference modes.

The problem formulation determines the model's main purpose. It is generally recommended that a modeller should not model 'the system', but 'the problem' (e.g., Sterman, 2000, p. 89). The idea is to stay away from trying to model the whole system and all its attributes, but to limit oneself to those parts and aspects that are relevant to the problem at hand and generate the problematic behaviour.

During the problem articulation phase, you need to determine the time horizon. In problems of socio-technical systems such as those studied at TU Delft, the focus is mostly on future-oriented system behaviour: the purpose of the model is to explore how the future may unfold under various circumstances. Therefore, the time horizon is selected as a time period ranging from the present to a point in the future. Frequently, a section of the past is covered as well, in order to simulate how the problem emerged. Nevertheless, there are SD applications that purely focus on explaining an existing problem, where the time scope lies only in the past.

Additionally, in this phase you may choose reference modes. These are defined as "a set of graphs and other descriptive data showing the development of the problem over time" (Sterman, 2000, p. 90). In the study of future-oriented developments, hypothesised behaviours can be used as references mode (Randers, 1980). For studies looking at past problems, existing data should be used for the reference modes.

In the **conceptualisation phase**, you sketch an outline of the model that is going to be constructed. Diagrams like Causal Loop Diagrams (CLDs), Stock Flow Diagrams (SFDs), Sub-System Diagrams (SSDs), and Bull's Eye Diagrams are frequently used to facilitate this. Later, once further research has been performed, updated versions of these same diagrams (especially CLDs, SFDs, and SSDs) can be used to communicate basic model structure. A dynamic hypothesis on how the structure may cause certain behaviour (e.g., the reference mode) to arise in the model is also part of the conceptualisation phase.

In the **formulation phase**, the SD model is specified: equations are defined and parameter values are given. This frequently starts with deciding which stock-flow structures need to be included in the model and where important delays play a role. There are different successful ways of approaching formulation. Two options are to model "inside out" or "outside-in". In "inside-out" modelling, you start with one of the more obvious stock-flow structures (e.g., the population submodel in a model on societal ageing, or the transmission sub-model in a model on an infectious disease) and gradually expand it. In "outside-in" modelling, you start with the data you want to use in the model, and start combining it bit by bit into model structures.

In the **evaluation phase**, the model is subjected to various tests to assess its quality. These tests respectively focus on whether the model has been *correctly* *constructed* (i.e., verification) and whether it is *fit for purpose* (i.e., validation) (Barlas, 1996; Hodges & Dewar, 1992; Oreskes, Shrader-Frechette, & Belitz, 1994). The term 'fit for purpose' in the validation process refers to testing whether the model fulfils the purpose as defined in the problem formulation. The concepts of verification and validation in SD correspond to a large extent to the way in which these terms are used in operations research and management science (Balci, 1994, 2013; Lane, 1995). The results of the tests can be communicated to convince stakeholders, policy-makers or other users of the model's quality, given the purpose. Once the model passes the verification and validation tests adequately, one or more base runs are selected for analysis and communication.

In the **policy testing phase**, the efficacy of policy levers is tested in an experimental set-up. In this phase, the modeller tests what will happen to the system behaviour if the model's structure is changed. Policy testing is not predictive, but rather an exploration of potential futures; it may consider uncertainties and scenarios in the evaluation of policy efficacy. Thus, this phase explores which policies may offer a promising or robust solution to the problem formulated at the outset.

Exercises

2. The first modelling cycle

2.1 Theory exercises

Exercise 2.1. Research problem

Note: This type of exercise builds on the theory on archetypes in the chapter on Problem Formulation and Conceptualisation (section 6.2).

Burn outs among university students – Explain why it can be useful to use an SD model when researching the causes for work pressure and burn outs amongst university students (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7459661/). Name exactly 2 examples of different system characteristics related to this research, including feedback.

Exercise 2.2. SFD to CLD

Note: This type of exercise builds on the theory on aggregated and disaggregated models (Section 6.4.5).

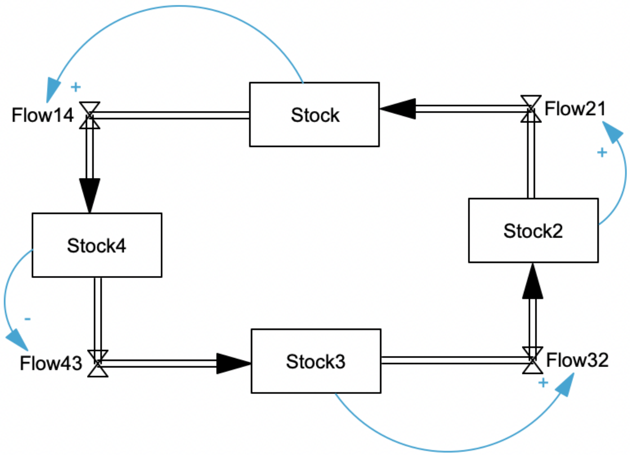


Figure 2.1

a. Draw a disaggregated CLD.

b. How many feedback loops are there in the SFD above?

Exercise 2.3. SFD to CLD

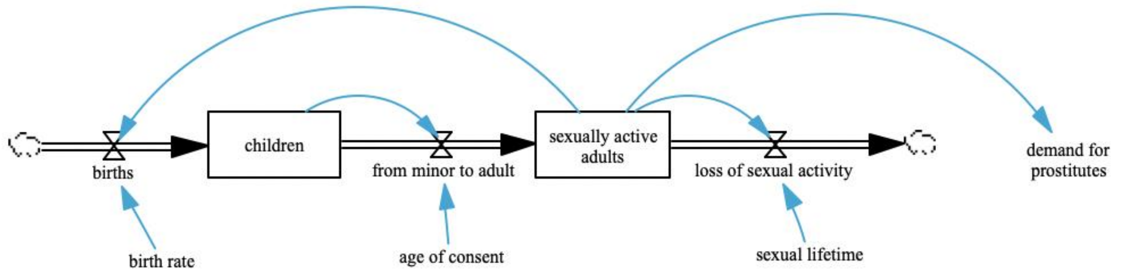


Figure 2.2. SFD of human trafficking

Consider the SFD of the population sub-model of a model related to prostitution and human trafficking displayed above.

a. Draw a disaggregated CLD.

b. How many feedback loops are there in the SFD above?

Exercise 2.4. SFD to CLD

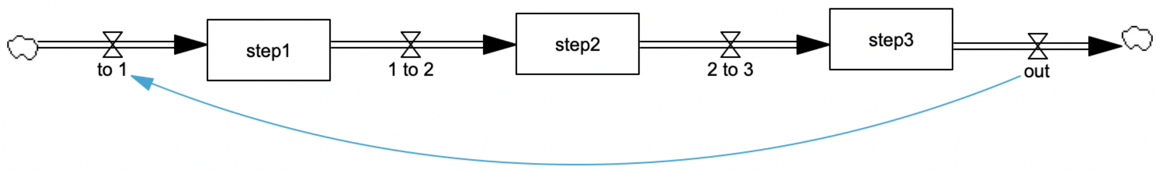


Figure 2.3

a. Draw a disaggregated CLD.

b. How many feedback loops are there in the SFD above?

Exercise 2.5. SFD to CLD

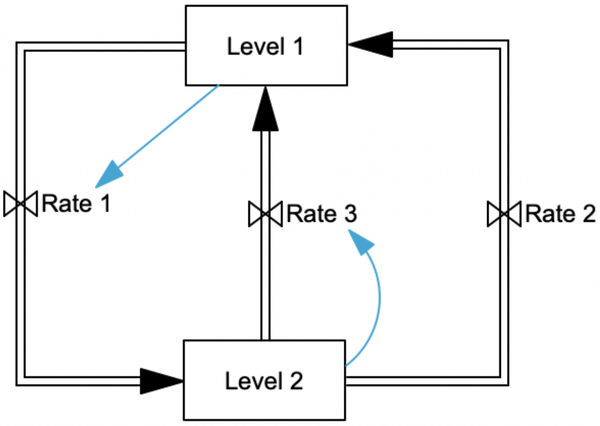


Figure 2.4

a. Draw a disaggregated CLD.
b. How many feedback loops are there in the SFD above?

Exercise 2.6. SFD to CLD

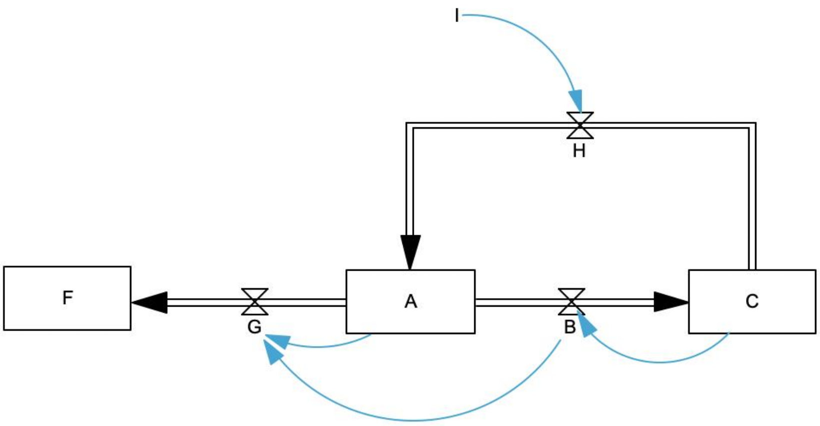


Figure 2.5

a. Draw a disaggregated CLD.
b. How many feedback loops are there in the SFD above?

Exercise 2.7. SFD to CLD

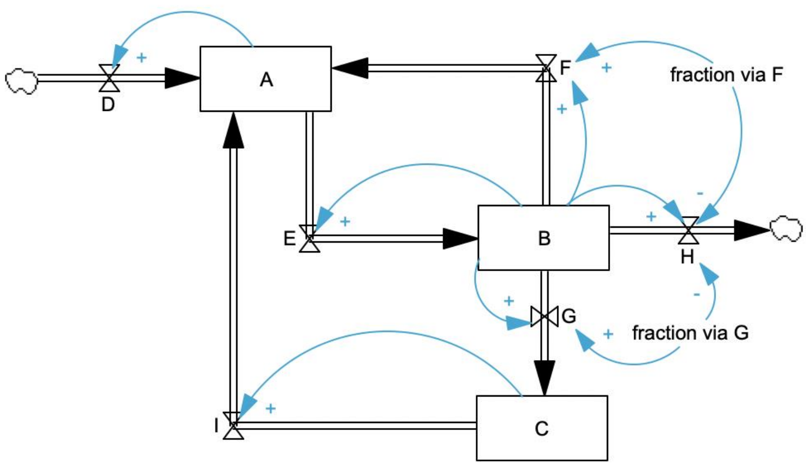


Figure 2.6

a. Draw a disaggregated CLD.
b. How many feedback loops are there in the SFD above?

Exercise 2.8. SFD to CLD

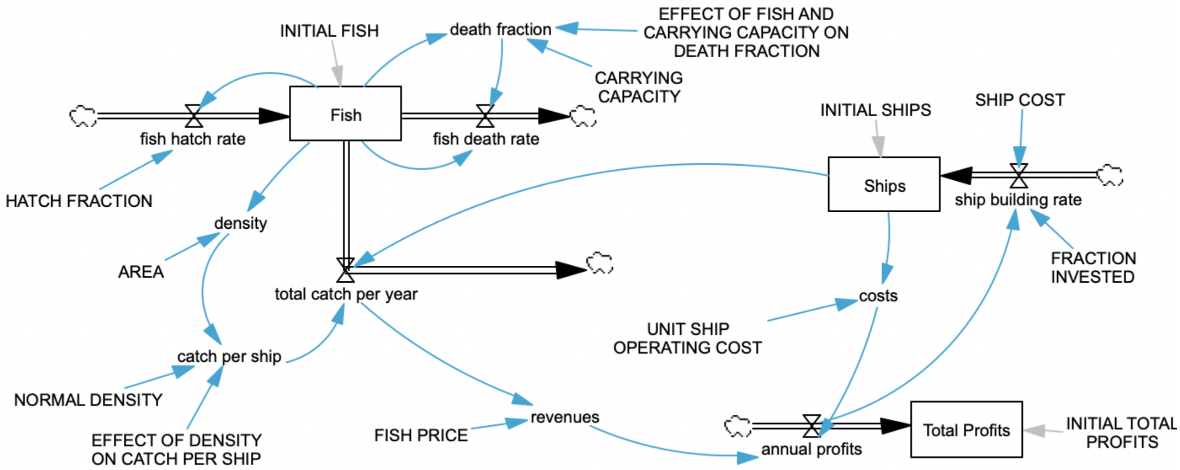


Figure 2.7

a. Draw a disaggregated CLD.
b. How many feedback loops are there in the SFD above?

Exercise 2.9. Units

The unit of time in a model concerning the large-scale introduction of electrical vehicles (EVs) is expressed in *month*. The production capacity of a company that produces EVs is modelled as a stock variable with units expressed in *EV/month*. The enormous growth of the expected demand for new EVs leads to an increase of the production capacity of EVs.

a. What unit needs to be used for this increase of the production capacity?

Exercise 2.10. Units

The unit of time in a model concerning the large-scale introduction of wind turbines is expressed in *months*. The production capacity of a company that produces wind turbines is modelled as a stock variable with units expressed in *turbine/month*. The enormous growth of the demand for new wind turbines leads to an increase of the production capacity of the company.

a. Which unit needs to be used for this increase of the production capacity?

2.2 Modelling exercises

Exercise 2.11. Cocaine

Example

The monthly change in the *total quantity of cocaine in a country* depends on the monthly quantity of *cocaine imports*, the monthly quantity of *cocaine used* and the monthly quantity of *cocaine confiscated* by the police. In a highly simplified model, we assume that 3000 kg of cocaine is used per month. The import of cocaine is constant and amounts to 4000 kg per month. And the monthly quantity of cocaine confiscated by the police is equal to 10% of the cocaine in the country. Suppose there was an initial amount of 3000 kg of cocaine in the country.

a. Make a *causal loop diagram* (CLD) of this problem. What behaviour do you expect?
b. Make a SD simulation model of this system. How would the *total quantity of cocaine in the country* evolve over time if initially there was 3000 kg of cocaine in the country? How would the *total quantity of cocaine in the country* evolve over time if initially there was 20000 kg of cocaine in the country? And how would the *total quantity of cocaine in the country* evolve over time if initially there was 10000 kg of cocaine in the country?

Exercise 2.12. Muskrat Plague

Example

Suppose there is a muskrat plague in a particular area. At first, there were 100 *muskrats*. The *New muskats* increase the number of *Muskrats* due to the autonomous increase in the number of muskrats per muskrat per year amounts to an average of 20 muskrats per muskrat per year. *Muskrats* decrease by *Muskrats* *caught*. Suppose that each year, 10 licences are granted to set muskrat traps. The licences are only valid for one year and each person holding a licence may set 10 traps. Assume the number of *muskrats caught per trap* is proportional to the number of muskrats and a *catch rate per trap* which is close to 0.2, say, on average 0.2, minimally 0.195, and maximally 0.205.

a. Draw a CLD of the problem.
b. Represent the system in Vensim. Simulate the model three different times with proportionality factors 0.195, 0.200 and 0.205, and make a graph of the muskrat population over a period of 10 years.
c. Compare your model structure and behaviour with the example model.
d. See what happens for different values of the proportionality factor and try to explain this.
e. Design and test an open-loop policy (see section 9.1.1).
f. Given the uncertainty (of the proportionality factor, of the initial number of muskrats, and of the exact number of muskrats at any time), design a better – more specifically a dynamic closed-loop (see section 9.1.2) – licensing strategy that allows for stabilization of the muskrat population. Modify your initial simulation model, and compare the outcome of your dynamic closed-loop policy with the outcomes of the open-loop policy.

Exercise 2.13. The Threat of the Feral Pig Example

Many American Departments of Natural Resources have adopted the position that feral pigs (Latin: *Sus scrofa*) are exotic, non-native wild animals that pose significant threats to both the environment and to agricultural operations. See, for example, the website of the Southeastern Wisconsin Invasive Species Consortium, on which information this exercise is partly based. Due to feral pigs' trampling and rooting behaviours, many American wildlife biologists and institutions are becoming increasingly concerned about the devastation these 'exotic' animals can cause to ecologically sensitive native habitats, particularly native plants and rare, threatened or endangered species.

Suppose one of the Departments hired you to develop a SD simulation model of the feral pig population to assess the dynamics of the population over time, given the departmental gaming rules. In this region, feral pigs may be removed at any time throughout the year, as long as those hunting them possess a valid licence. Each year, 10 *licences* are granted to set feral pig *traps*. The licences are only valid for one year. Each person holding a licence may set 10 *traps*. The number of *feral pigs caught per trap* is proportional to 15% to 17% of the total number of feral pigs.

Assume that, at this moment, there are about 20000 feral pigs in the area, that feral pigs can mate at any time of the year, that sows produce on average 4 litters per year with each litter consisting of 8 piglets on average, and that half of the pig population consists of sows.

a. Draw a simple CLD of the feral pig problem.
b. Make a simple SD simulation model of the feral pig population to assess the dynamics of the population over time, given the departmental gaming rules.
c. Make a graph of the number of feral pigs over time. Show what happens in the graph for different values of the proportionality factor (15%, 16% and 17%).
d. Try to verify and validate your model. What would you do?
e. Given the uncertainty (of the proportionality factor, of the initial number of feral pigs, and of the number of new pigs), design a better (dynamic) licensing strategy that allows for stabilisation of the feral pig population at less than 25000 animals.

3. Model formulation 1

3.1 Theory exercises

Exercise 3.1. Units

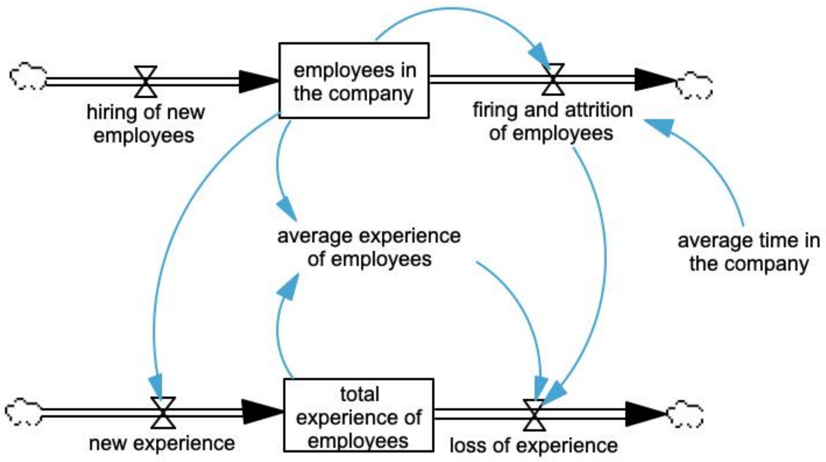


Figure 3.1 Sub-model

Consider the fully displayed sub-model of a company with a large number of employees (Figure 3.1). The model is about gain and loss of experience, which is measured in weeks. The SD model aggregates the experience of all employees in the company. All relevant variables are displayed; no constants or parameters are used that are not shown.

Employees gain experience over the course of time. If an employee leaves the company, it is assumed that the employee leaves with his/her experience, which equals the average experience. The formula for *loss of experience* is equal to the *firing and attrition of employees* times the *average experience of employees*. The formula for *average experience* is equal to the *total experience of employees* divided by the number of *employees in the company*. The *number of employees* is expressed in 'person', the unit of time is expressed in 'week'.

a. What units should be used for *the average experience of employees* and the *total experience of employees*?

Exercise 3.2. SFD to CLD

Various elements of the Dutch housing market, including the social housing market, have been criticised by different parties for the past few years. According to the European Commission, the rules and organization of the Dutch corporation system did not correspond to the purpose of social housing. That is, the social rental market was not restricted to the poor. Although nobody complained, the European

Commission argued that the Dutch rental market did not comply with the rules of free competition, creating a situation of unfair competition with the private rental market, resulting in a distorted housing market. After long negotiations, the Dutch government and the European Commission agreed that after 1/1/2011, 90% of all social housing vacancies would be allocated to families with incomes up to €28,475 per year – which is below the modal income. However, no agreement was reached about the majority of above-modal income families already living in social housing. This resulted in a situation in which families with an income above the modal were not allowed to move house within the social housing sector any more. For them, moving meant, from then on, having to leave the social housing sector and buy or rent on the (excessively expensive) private market.

Before 1/1/2011, the housing market was a pull system: after an attractive house had become vacant, another social housing family would soon move into the vacant house, freeing up their own house, which would soon be filled up by yet another social housing family, until finally, a 'starter house' would become vacant and be occupied by young entrants into the social housing system. Due to the new regulations, higher-income families now stay put, which has brought the dynamics on the social renting market to a halt.

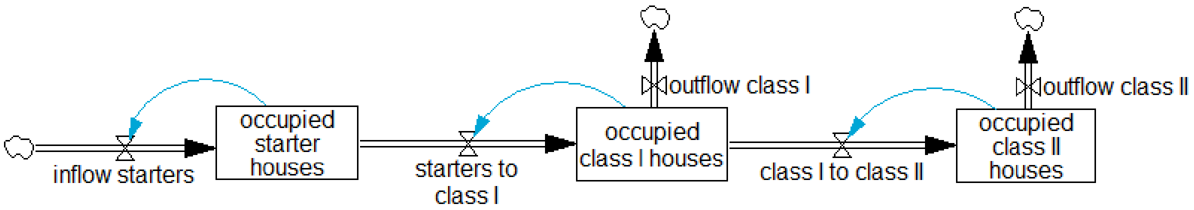


Figure 3.2. Social housing simulation model

Consider the SFD of a Dutch social housing simulation model displayed above (Figure 3.2).

a. Draw a disaggregated CLD.

b. Draw an aggregated CLD.

Exercise 3.3. CLD to SFD

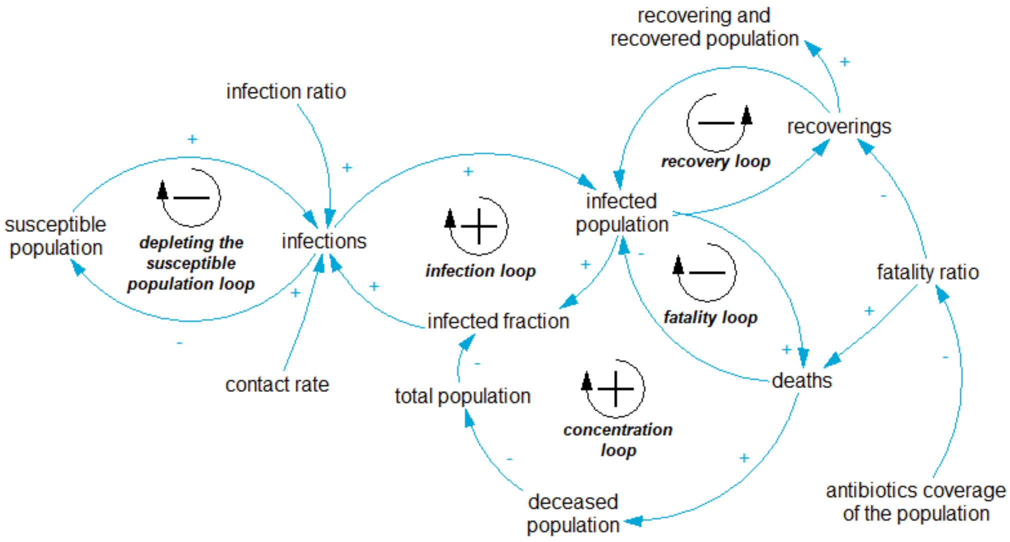


Figure 3.3. CLD of disease outbreak

Consider the detailed CLD on the outbreak of a disease displayed above (Figure 3.3).

a. What should be the stocks in the model to be made following this CLD?

b. Draw a SFD which corresponds to this CLD.

Exercise 3.4. Archetype

Note: This exercise builds on the theory on archetypes in the chapter on Problem Formulation and Conceptualisation (section 6.6.2).

Drought – Overuse of surface water and groundwater reduces the amount and quality of the water supply. When actors pump too much groundwater or surface water, the resources is depleted before it can be replenished. As the water table lowers, lakes, rivers, streams and canals that are connected to the groundwater have less supply to pull from. Additionally, especially along the coasts, excessive pumping ruins water quality by salt water intrusions.

a. Which archetype fits this case?

Exercise 3.5. Delays

Note: This exercise builds on the theory on delays in the chapter on formulation (section 7.4.2). Make sure you made the modelling exercises from Exercise 3.7 till Exercise 3.11 before doing this exercise.

Consider the figure below Figure 3.4.

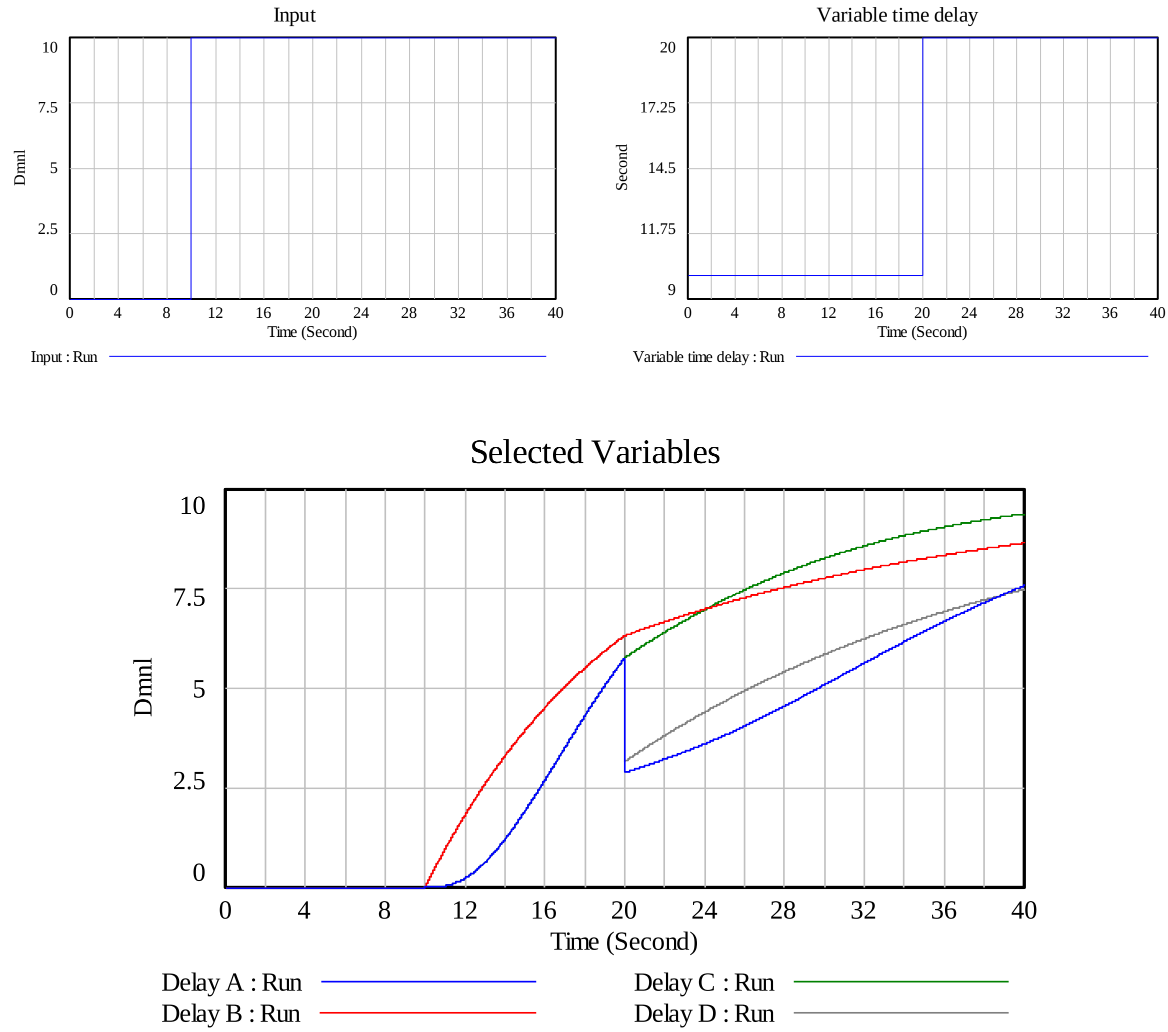


Figure 3.4. Input (*above left*), variable time delay (*above right*) and different delays of the input with the variable time delay (*below*).

a. What are the delay types (material or information) and orders (1st, 3rd, or FIXED) of each run?

Delay	Type	Order
Delay A		
Delay B		
Delay C		
Delay D		

3.2 Modelling exercises

Exercise 3.6. Pneumonic Plague Example

Introduction

On 3 August 2009, media reported on an outbreak of pneumonic plague in north-west China ("Second plague death in west China," 2009):

> *“A second man has died of pneumonic plague in a remote part of north-west China where a town of more than 10,000 people has been sealed off. […] Local officials in north-western China have told the BBC that the situation is under control, and that schools and offices are open as usual. But to prevent the plague [from] spreading, the authorities have sealed off Ziketan, which has some 10,000 residents. About 10 other people inside the town have so far contracted the disease, according to state media. No-one is being allowed [to] leave the area, and the authorities are trying to track down people who had contact with the men who died. […] According to the WHO, pneumonic plague is the most virulent and least common form of plague. It is caused by the same bacteria that occur in bubonic plague – the Black Death that killed an estimated 25 million people in Europe during the Middle Ages. But while bubonic plague is usually transmitted by flea bites and can be treated with antibiotics, [pneumonic plague, which attacks the lungs, can spread from person to person or from animals to people], is easier to contract and if untreated, has a very high case-fatality ratio.”*

Modelling a Pneumonic Plague Outbreak

You are asked to make a SD model of this outbreak. Use the following assumptions: The *total population* of Ziketan amounted initially to 10000 citizens. New *infections* make that citizens belonging to the *susceptible population* part of the *infected population*, which initially consists of just 1 person. The number of *infections* equals the product of the *infection ratio*, the *contact rate*, the *susceptible population*, and the *infected fraction*. Initially, the *normal contact rate* amounts to 50 contacts per week and the *infection ratio* to a staggering 75% per contact. The *infected fraction* equals the *infected population* over the sum of all other subpopulations. If citizens from the *infected population* die, they enter the statistics of the *deceased population*; otherwise they are quarantined to recover. The *recovering* could be modelled simplistically as $(1 - fatality ratio) * infected\ population/recovery\ time$.

Suppose for the sake of simplicity that the average *recovery time* and the *average decease time* are both 2 days. The *fatality ratio* depends on the *antibiotics coverage of the population* which – in this poor part of China – is 0% at first. The fatality ratio is 90% at 0% *antibiotics coverage of the population* and 15% at 100% *antibiotics coverage of the population*. Assume for the sake of simplicity that those belonging to the *recovering population* do not pose any threat of infection, either because they are fully quarantined or because they are not contagious any more.

Simulate the model using a time horizon of 30 days.

a. Make a SD simulation model of the local pneumonic plague epidemic.
b. Assume in your base case simulation that the *antibiotics coverage of the population* remains 0%. Make graphs of the evolution of the *infections*, the *deaths*, the *recovering population* and the *deceased population*.
c. Assume now that *antibiotics coverage of the population* is 100%. Compare the evolution of the *infections*, the *deaths*, the *recovering population*, and the *deceased population* with those obtained in the previous question.
d. Make a detailed and highly-aggregated CLD of this model.
e. What could stop the epidemic? Explain based on the structure of the model.

Exercise 3.7. Step, Ramp, Time, Sin

This technical exercise allows you to get familiar with some pre-defined functions.

a. Open a new model. Change the model settings so that the model simulates from 0 to 20 years with a time step of 0.0625. Use the Euler numerical integration method.
b. Add a variable named *cte*. Give it a constant value 5 and dimensionless units (Dmnl). Simulate the model. Select the variable and visualize its behaviour with the graph tool.
c. Add a variable named *step up*. Add a STEP function to it which forces a discrete step with 6 units at time 10, i.e. $STEP(6,10)$, using units Dmnl. Simulate the model. Select the variable and visualize its behaviour with the graph tool.
d. Add a variable named *step down*. Add the function $10 - STEP(6,5) - STEP(3,15)$ to it, using units Dmnl. Simulate the model. Select the variable and visualize its behaviour with the graph tool.

e. Add a variable named *ramp up*. Add a RAMP function to it that grows from time 10 to time 20 with a slope of 1 per year, using units Dmnl. Select the variable and visualize its behaviour with the graph tool.

f. Use the shadow variable tool to add the predefined variable *Time*. Select the variable and visualize its behaviour with the graph tool.

g. Add a variable named *sine* with units Dmnl. Add a SIN function to it as a function of *Time*. In other words, link a *Time* shadow variable to the sine variable, open the equation editor and add the equation: $5 + 2 * sin(Time)$. Select the variable and visualize its behaviour with the graph tool.

h. Now select all auxiliaries (hold the shift button while clicking them) and visualize them all, except for the *Time* shadow variable.

Exercise 3.8. Min, Max, MinMax, MaxMin

This technical exercise helps you to get familiar with different Max and Min constructs. Use the previous model from Exercise 3.7 for this exercise.

a. Add a variable named *max cte and step up*. Open the equation editor and add the following function: $MAX(cte, stepup)$. Simulate the model. Select the three variables (*cte*, *step up* and *max cte and step up*) and visualize their behaviour with the graph tool. What does the MAX function do?

b. Now build a function that takes the maximum of all variables from Exercise 3.7. Note: since max functions can only take two arguments, you will need to build nested max functions.

c. Add a variable named *min step down and sine*. Open the equation editor and add the following function: $MIN(stepdown, sine)$. Simulate the model. Select the three variables (*step down*, *sine*, *min step down and sine*) and visualize their behaviour with the graph tool. What does the MIN function do?

d. Now build a function that takes the minimum of all variables from Exercise 3.7. Note: since min functions can only take two arguments, you will need to build nested min functions.

e. Add two constants named *floor* and *ceiling* respectively. Set the *floor* to 4 and the *ceiling* to 6. Add three auxiliary variables called *sine with ceiling*, *sine with floor* and *truncated sine*. Suppose that the *sine with ceiling* equals $MIN(sine, ceiling)$, that the *sine with floor* equals $MAX(sine, floor)$, and that the *truncated sine* equals $MAX(MIN(sine, ceiling), floor)$. Simulate the model and visualize the behaviours of these three variables.

f. Draw you conclusion: what may the MIN and MAX functions be used for?

Exercise 3.9. Stocks

This technical exercise helps you to get familiar with the effects of stock variables. It builds on the previous two exercises. Use the previous model from Exercise 3.8 for this exercise.

a. Add a stock variable named *Material1* with initial value of 0. Connect an inflow *Material in* and an outflow *Material out* to this stock. The outflow *Material out* equals the stock variable divided by a *depletion time* of 1 year. Set the *Material in* inflow equal to each of the variables in Exercise : *cte*, *step up*, *step down*, *ramp up*, *Time* and *sine*. Compare for each variable the behaviour of the inflow and outflow of the stock. What happens?
b. Do the same for the variables added in Exercise 3.8. Set the *Material in* inflow equal to each of the variables in Exercise 3.8: *max cte and step up*, *min step down and sine*, *sine with ceiling*, *sine with floor* and *truncated sine*. Compare for each variable the behaviour of the inflow and outflow of the stock. What happens?
c. Add a stock variable named *Cumulative in* with initial value equal to 0. Add an inflow called *Cumulative inflow*. Equate the inflow to each of the variables in Exercise : *cte, step up, step down, ramp up, Time* and *sine*. Compare for each variable the behaviour of the inflow and stock variable. What happens?
d. Add a stock variable named *Cumulative out* with initial value of 100. Add an outflow called *Cumulative outflow*. Equate the inflow to each of the variables in Exercise : *cte, step up, step down, ramp up, Time*, and *sine*. Compare for each variable the behaviour of the outflow and stock variable. What happens?
e. Draw your conclusion: what is special about stock variables?

Exercise 3.10. First-Order Material & Information Delays

This technical exercise builds on Exercise 3.9. It helps you to get familiar with first-order material delays and first-order information delays.

a. The stock with an inflow and outflow proportional to the stock in Exercise 3.9 is in fact a first-order material delay structure: the inflow is the input and the outflow gives the response or output of the delay. Now, let’s build a first-order information delay: add another stock variable *Information1* with initial value of 0, an inflow *Information1In*, an auxiliary *In1*, an auxiliary *Out1*, and a *Delay*

Time of 1 year. The auxiliary *Out1* should be equal to the stock variable *Information1*, the inflow *Information1In* should be equal to: $(1 - Information1)/Delay\,Time$. Replace the variable *Depletion Time* in the first-order material delay by the *Delay Time*. Equate the auxiliary *In1* to each of the variables in Exercise : *cte*, *step up*, *step down*, *ramp up*, *Time*, and *sine*. Simulate and compare the inputs and outputs of this first-order information delay with the first-order material delay. Draw your conclusion: what does the information delay do? Do the responses of the material delay and information delay differ?

b. Now change the constant *Delay Time* into the following function over time: $1 + STEP(3{,}5) - STEP(2{,}10) - RAMP(0.1{,}15{,}20)$. Compare the effect on the first-order material delay and the first-order information delay. What happens? Do the responses of the material delay and information delay differ now? What is your conclusion?

c. Why do you think the terms 'material delay' and 'information delay' are used? What could material and information delays be used for?

Exercise 3.11. Higher-Order Delays

This technical exercise builds on Exercise 3.10. It helps you to get familiar with higher-order material and information delays. This exercise allows you to simulate and analyse the consequences of different delays (information and material), of different orders (1^{st} order, 3^{rd} order, 10^{th} order, and pipeline delays aka delay fixed), for different inputs, and for fixed and variable delay times.

a. Select the material delay structure, the information delay structure, and the *step up* variable from the model developed in Exercise 3.7. Copy them, open a new model, paste them, and adapt your model settings. Change the *delay time* into a constant 1 year. Name the simulation run, for example, '1st order', and simulate the model.

b. Extend both structures to a second-order delay by copying these structures linking the output variables to the input values and adding a variable *delay order* and *delay time divided by delay order*. The formula of the variable *delay time divided by delay order* is of course equal to the *delay time* divided by the *delay order*. Rename the simulation run, for example, '2nd Order', set the *delay order* equal to 2, and simulate the model again. Visualize and compare the 1^{st} order and 2^{nd} order outputs of the material delay and the information delay. What is the difference between the 1^{st} order and 2^{nd} order delays? What is the difference between material delays and information delays?

c. Do the same for a 3rd order delay: extend the stock-flow structures, change the order, change the run name, simulate, and compare their behaviours.

d. Delays can also be specified with predefined functions. Make a new variable called *MaterialDelayFunction* and a new variable called *InformationDelayFunction*. Open these variables with the equation editor and select the function DELAY3I for the *MaterialDelayFunction* and SMOOTH3I for the *InformationDelayFunction*. Select the same input *In*, *delay time*, and initial value 0 as before. Simulate the model and compare the outputs of the *MaterialDelayFunction* and the 3rd order material delay stock-flow structure. Do the same for the outputs of the *InformationDelayFunction* and the 3rd order material delay stock-flow structure.

e. Now change the constant *delay time* into the following function over time: $1 + STEP(3,5) - STEP(2,10) - RAMP(0.1,15,20)$. Compare the effect on the 1st order material delay and the 1st order information delay. What happens? What could be concluded?

f. Change the 3rd order delay function and 3rd order smooth function into DELAY N and SMOOTH N which can be selected in the 'all' functions class. These functions do the same, but now the order can to be set as well. Choose for example a 10th order. Rename the run, simulate it, and compare the response. Double the order, rename the run, simulate it, and compare the output. What conclusion can you draw from this?

g. Add another auxiliary variable *MaterialDelayFixed*, specify its function as DELAY FIXED (Vensim) or DELAY (Stella) and simulate it. Compare the output to the highest-order material delay function. What conclusion can you draw from this: what is a delay fixed aka pipeline delay and what does it do?

h. Copy and paste the other inputs from the model developed in Exercise 3.7, connect them, and compare the responses of the delay and smooth structures with the corresponding inputs.

i. What are your overall conclusions regarding material and information delays? Why do you think the terms 'material delay' and 'information delay' are used?

Exercise 3.12. Lookups, With Lookups, Time Series Example

If you solved Exercise 3.6 on the outbreak of pneumonic plague in China, then open your model, and set the antibiotics coverage of the population back to 0%. If you have not solved Exercise 3.6 yet, then read the description and open the corresponding model here (be aware that the model in the link is not complete, so compare it with the description in Exercise 3.6 and complete it).

a. The fatality ratio is 90% at 0% *antibiotics coverage of the population* and 15% at 100% *antibiotics coverage of the population*. Add a table function (WITH LOOKUP or LOOKUP in Vensim) named *effect of antibiotics coverage on fatality ratio* to include this effect.

b. The outbreak of an extremely contagious deadly illness such as pneumonic plague actually causes the *contact rate* to drop, because people get ill or because they isolate themselves. Adapt the model by closing the 'loop' between the *infected fraction* and the *contact rate*. Create a function *impact of the infected fraction on the contact rate* that, multiplied with the *normal contact rate* (without illness and isolation), gives the effective contact rate. The function takes a value of 1 at an *infected fraction* of 0%, of 0.5 at an *infected fraction* of 10%, of 0.25 at an *infected fraction* of 20%, of 0.125 at an *infected fraction* of 30%, of 0.0625 at an *infected fraction* of 40%, of 0.03125 at an *infected fraction* of 50%, and so on. Simulate the model over a time span of 1 month. Make graphs of the evolution of the *infections*, the *deaths*, the *recovering population*, and the *deceased population*. Does this reduction of the natural contact rate result in the desired effect?

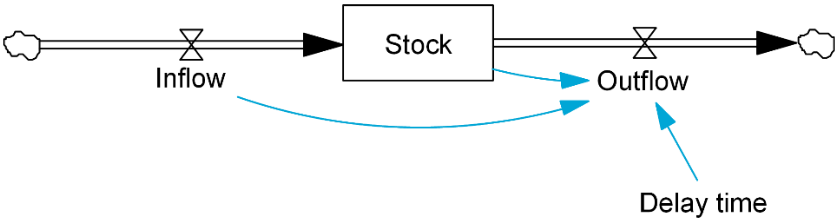
c. Suppose that the *antibiotics coverage of the population* increases linearly in de first week of the epidemics from 0% to 100%. Add a table function to simulate this effect. What is the consequence for the variable *deceased population*?

4. Model formulation 2

4.1 Theory exercises

Exercise 4.1. SFD to CLD

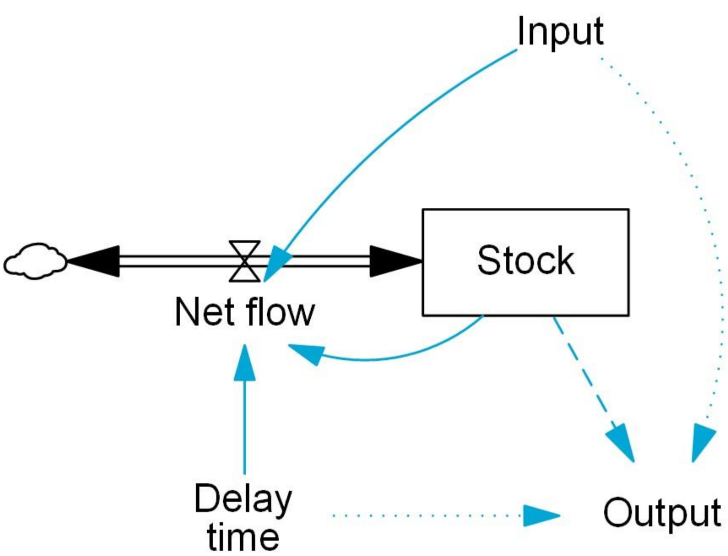
Consider the SFD below. The *Outflow* is defined as *DELAY3I (Inflow, Delay time , Stock / Delay time)*.



a. Draw the disaggregated CLD of this SFD. Make the implicit delay function explicit and exclude initial causes.
b. Draw the aggregated CLD of this SFD.
c. Explain how the *Outflow* changes at the moment the *Delay time* is halved.

Exercise 4.2. SFD to CLD

Consider the SFD below. The *Net flow* is defined as (*Input – Stock*) / *Delay time.*



First, consider the situation in which the *Output* is defined as *Stock* (i.e., dashed line*)*.

a. Draw the disaggregated CLD of this diagram. Make the implicit delay function explicit.
b. Draw the aggregated CLD of this SFD.

Second, consider the situation in which the *Output* is defined as *SMOOTH3I* (*Input* , *Delay time* , *Stock*) (i.e., dotted line).

c. Draw the disaggregated CLD of this diagram. Make the implicit delay function explicit. Make the implicit delay function explicit and exclude initial causes.
d. Draw the aggregated CLD of this SFD.
e. Explain how the *Outflow* changes at the moment the *Delay time* is halved.

Exercise 4.3. Archetype

Biodiversity – Despite decade-long attempts to combat loss of biodiversity, Earth is losing animal and plant species rapidly. Why can't we seem to combat this? The marine biologist Daniel Pauly has offered an explanation for this. He reasons that each generation fish researchers accepts the situation at the beginning of their careers as a baseline. Fish stocks, and the number of alive fish species at that moment, count as the baseline. Subsequently, that baseline is used to determine fish quota. These fish quota are generally not complied with by countries. Fish stocks decrease. And a next generation researchers accept the newer, poorer situation as a new norm. This is how the baseline shifts over generations.

a. What archetype could you recognize in this case? Name the archetype and explain it in 1 sentence

Exercise 4.4. Archetype

The birth month effect – The Royal Dutch Association for Football (KNVB) is rethinking their selection policy for talented players in the youth division. In the old system, players were selected based on their age category: the best players of a birth year were selected into the talented selections. Sport clubs invest more time and resources in the top selections. Research has shown the older players born in January, who are further in their cognitive and physical development than those born in December of the same calendar year, are more likely to be selected in those talented teams. As a result of more and better training, the relatively older players developed faster than their un-selected peers. Consequently, football clubs had significantly more players born early in the year as opposed to later in the year. They were losing out on talent.

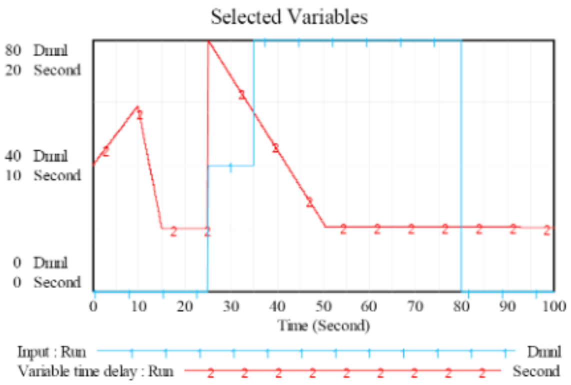
a. What archetype could you recognize in this case? Name the archetype and explain it in 1 sentence

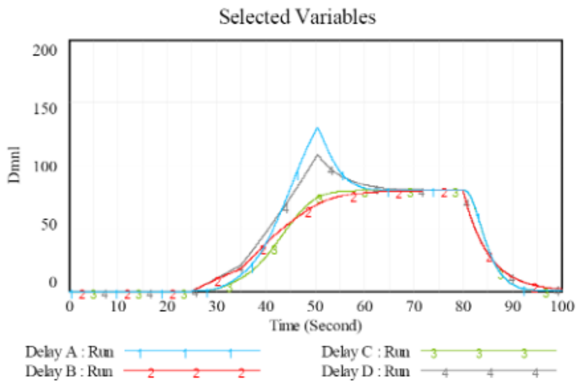
Exercise 4.5. Research problem

Antibiotics – Explain why it can be useful to use an SD model when researching agricultural and medical use of antibiotics in the European Union. Name exactly 2 examples of different system characteristics related to this research, including feedback.

Exercise 4.6. Delays

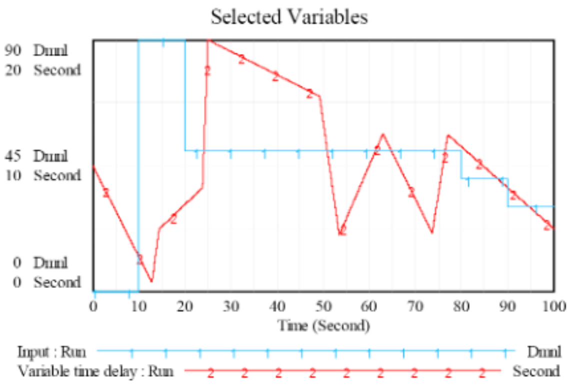
What type of delays (material or information) and order (1st, 3rd, or FIXED) of delay 1, 2, 3, and 4?

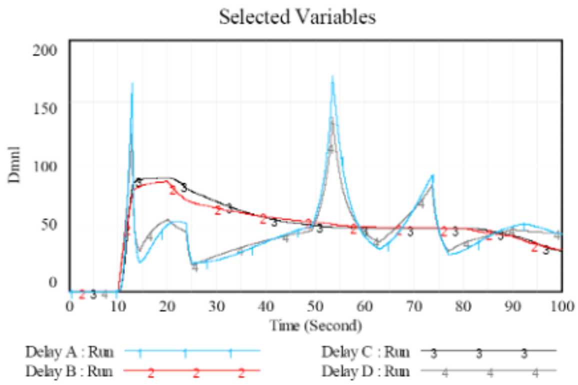




Exercise 4.7. Delays

What type of delays (material or information) and order (1st, 3rd, or FIXED) of delay 1, 2, 3, and 4?





4.2 Modelling exercises

Exercise 4.8. Collapse of Civilisations Example

This case is to a very large extent based on case 4.3 from Martín García (2006) (reproduced with permission). The introduction and the model upon which the remainder of the case is based is highly similar to Juan Martín García's model.

Introduction

One of the great mysteries of human history has been the sudden collapse – around 800 AD – of one of the main centres of Mayan civilization in Central America, at a time when it was apparently peaking in terms of culture, architecture and population. Research suggests that the population increased – just before the collapse – to about 200 to 500 persons per square kilometre. This population density required advanced agriculture and/or large-scale trade. Within two to four Mayan generations, the population density dropped to less than 20 persons per square kilometre (i.e., the same as 2000 years earlier). Furthermore, after the collapse, whole areas remained almost uninhabited for some thousand years. Some of the environmental changes appear to have been as long-lasting as the loss of population. Lakes that were apparently centres of settlement in the Maya era have not yet entirely recovered in terms of productivity.

No one knows exactly why this society of several million people collapsed, but new research shows a gradually tightening squeeze between population and environment that may have been crucial to the fall. Scientists[1] estimate that exponential growth in Mayan population took place for at least 1700 years in the tropical lowlands of what is now Guatemala, such that the population doubled every 408 years. This trend may have caught the Maya in a strange trap. Their numbers grew at a steadily increasing pace, but, for many centuries, the growth was too slow for any single generation to see what was happening. Over the centuries, the increasing pressure on the environment may have become impossible to sustain. Yet the squeeze could have been imperceptible until the final population explosion, just before the collapse.

New estimates for the population of the southern lowlands are based largely on a detailed survey of traces of residential structures that were built, occupied and abandoned over the centuries. The studies focus on the region of two adjacent lakes

[1] Dr. Don S. Rice, assistant professor of archaeology at the University of Chicago and adjunct assistant curator of archaeology at the university museum, Dr. Edward S. Deevey, leader of the research team and graduate research, curator of paleoecology at the Florida State Museum, H.H. Vaughan, Mark Brenner and M.S. Flannery of the University of Florida and Prudence M. Rice, assistant curator of archaeology and assistant professor at Florida University.

(Lake Yaxha and Lake Sacnab) in the Peten lake district of northern Guatemala. The area was inhabited as early as 3000 years ago and the first agricultural settlements appeared there about 1000 BC. The land was largely deforested by 250 AD. The gradual intensification of agriculture and increasing number of settlements seem to have caused severe cumulative damage to an originally verdant environment. Essential nutrients washed away into the lakes, diminishing the fertility of agricultural land. Increases in phosphorus in the lakes from agriculture and human waste seem to have aggravated the environmental damage. Tropical environments are notoriously fragile, so that at one point, the population became too large for the environment to sustain.

Simulate this SD model over a period of 2000 years from 1000 BCE (-1000) till 1000 CE.

Model description

At the time of onset of agricultural developments, around 100000 persons lived in the area. The *Population* saw a *Natural increase* due to a *Natural increase rate* of 0.17% per year. The *Population* could also decrease due to *Emigration*. For now, assume that this flow is equal to 0. We assume the *Consumed food per person* was around 400 kg per person per year, which roughly corresponds to a bit more than one kg per day. Define the *Food demand* using the *Population* and the *Consumed food per person*.

The Mayans converted *Forest* into agricultural *Lands* by *Deforestation*. The *Forest* initially measured 4992 km^2, and the *Lands* used 8 km^2, making the total area 5000 km^2. Assume for now that *Deforestation* is 0 and pick a suitable unit for this variable.

The *Fertility of lands*, initially equal to 5 million kg of food produced per square km, decreased due to *Losses in fertility*. There was a *Potential fertility to be lost*, equal to the difference between the *Fertility of lands* and the *Minimum fertility*. This flow is the product of the *Potential fertility to be lost* and the lowest value of either the *Max relative fertility reduction* (equal to 2), or the quotient of *Lands* and *Forest* to the power of *Fertility loss exponent*, in which the exponent is equal to 1.9 to get the right population dynamics, divided by the *Intensity of agriculture*. The *Intensity of agriculture* is equal to 1 year to model the speed by which the *Fertility of lands* changes.

The *Food produced* is the product of *Lands* and *Fertility of lands*. Define the *Emigration* flow now as the *Gap in food production* divided by the *Consumed food per person*, times an *Emigration ratio* of 5% per year, as due to food redistribution only a small proportion of people actually left.

The Mayans deforested the area they needed to close the *Gap in food production* (in kg) between the *Food produced* and the *Food demand*, given the *Fertility of lands* (kg/km^2) and the *Intensity of agriculture*. *Deforestation* should, therefore, be defined as the minimum of either the *Gap in food production* divided by the *Fertility of lands*, or the area of *Forest* divided by the *Maximum deforestation factor*, and everything divided by the *Intensity of agriculture*. The *Minimum fertility* is equal to 1 kg per square kilometre per year, and the *Maximum deforestation factor* is 4. The *Gap in food production* is the difference between the *Food demand* and the *Food produced*.

Questions

Build the simulation model according to the above description and answer the following questions.

a. Create an CLD on a high aggregation level.
b. What are correct model settings? Give INITIAL TIME, FINAL TIME, TIME STEP and Integration type.
c. Generate a graph for the *Population*. Give a structural explanation of its behaviour, and refer to loops in the CLD.
d. What archetype fits the model behaviour? Explain why.
e. Perform a structural validation test on the model. Describe the name of the test, how you performed it, what you observe, and your conclusions.
f. Perform a behavioural validation test on the model. Describe the name of the test, how you performed it, what you observe, and your conclusions.
g. A student made this model with five mistakes (find the incorrect model in the example folder). Find the errors, give the variable names, and the five improved equations.

Exercise 4.9. Gangs and Arms Races

Example

Arms races are escalation processes between two (or more) nations or parties in a conflict that 'watch each other and [both] respond to [uncertain] arming activities of their opponent with [relatively greater] arming activities of their own' (Bossel, 2007, p. 36). The number of weapons held by one party may actually deter the other party from attacking and vice versa, resulting in a situation of escalating armed peace. This exercise is based on the escalation model described in Bossel (2007, Z507).

Suppose there are two gangs, gang A and gang B. Initially, the *arms stock of gang A* amounts to 100% of the weapons needed to destroy gang B, and the *arms stock of gang B* amounts to 100% of the weapons needed to destroy gang A. The *arms stock of gang A* only increases or decreases via the *arming of gang A*; the *arms stock of gang B* only increases or decreases via the *arming of gang B*. Suppose that the arming of both gangs depends on an autonomous arming rate due to their intrinsic interest in arms and arming – the *autonomous arming rate A* and *autonomous* *arming rate B* respectively – and on an arming rate relative to the expected first-order arming of the adversary, that is, the arming of gang B from the point of view of gang A without consideration of the arming of gang A and vice versa. This *relative arming rate of gang A* – in terms of the weapons needed to destroy gang B – then equals the product of the *overassessment factor of gang B arming by gang A*, the *arms obsolescence rate of gang A*, and the *arms stock of gang B* minus the *arms obsolescence rate of gang A* times the *arms stock of gang A*. The same applies to gang B: the *relative arming rate of gang B* – in terms of the weapons needed to destroy gang A – equals the product of the *overassessment factor of gang A arming by gang B*, the *arms obsolescence rate of gang B*, and the *arms stock of gang A* minus the *arms obsolescence rate of gang B* times the *arms stock of gang B*. Assume that the *autonomous arming rate of gang A* and the *autonomous arming rate of gang B* are both equal to 5% of the weapons needed to destroy the other gang per month and that the *arms obsolescence rates* of both gangs equal 10% of the weapons needed to destroy the other gang per month.

a. Suppose that gang A over-assesses the arming of gang B by 10% (i.e., the *overassessment factor of gang B arming by gang A* is 110%) and that gang B correctly assesses the arming of gang A (i.e., the *overassessment factor of gang A arming by gang B* is 100%). Model the arms race. What behaviour do you expect? Simulate the model over a period of 100 months. What behaviour does the simulation show? Why?

b. Suppose that gang A underestimates the arming of gang B by 50% (i.e., the *overassessment factor of gang B arming by gang A* is 50%), and that gang B correctly assesses the arming of gang A, (i.e., the *overassessment factor of gang A arming by gang B* is 100%). Change the parameter, rename the run, and simulate the model again over a period of 100 months. What behaviour do you expect and what behaviour does the simulation show? Why?

c. Make a CLD of arms races between two gangs. Use it to explain the link between structure and behaviour.

Exercise 4.10. Unintended Family Planning Benefits Example

Levitt and Dubner (2005, Chapter 4) argue in *Freakonomics* that dropping crime rates are in fact an “unintended benefit” of legalised abortion. In this exercise, we will build a simplistic, purely hypothetical, simulation model to simulate the first-order effects on crime statistics of a sudden drop in the birth rate of families with multiple problems – be it by voluntary abortion or by successful family planning measures.

We focus on families with multiple problems only, and assume that individuals born in families with multiple problems are indeed trapped, but that they do not necessarily resort to crime. Model an ageing chain of *kids, youngsters, adults* and *retirees.* Initially, there are 1 million *kids*, 1 million *youngsters*, 3 million *adults*, and 750000 *retirees* within these families with multiple problems. Suppose for the sake of simplicity that only *retirees* die, on average after an *average time as retiree* of 15 years, which means that *deaths* equals *retirees* divided by *average time as retiree*. Similarly, *adults* flow from *adults to retirees* after an *average time as adult* of 40 years, *youngsters* from *youngsters to adults* after an *average time as youngster* of 12 years, and *kids* from *kids to youngsters* after an *average time as kid* of 12 years. Both *adults* and *youngsters* give birth: the *birth* inflow is thus the sum of the *adults* times the *annual fertility rate of adults* of 3 percent and the number of *youngster*s times *the annual fertility rate of youngsters* of 0.3%.

Suppose 6 million crimes are committed annually by others, that is, by criminals that are not part of families with multiple problems. Apart from these *crimes by others*, *crimes* are committed by *criminal kids* at a rate of 2 *criminal acts per criminal kid* per year, by *criminal youngsters* at a rate of 4 *criminal acts per criminal youngster* per year, by *criminal adults* at a rate of 12 *criminal acts per criminal adult* per year, and *by criminal retirees* at a rate of 4 *criminal acts per criminal retiree* per year. Suppose that, in these families with multiple problems, the *percentage of kids with criminal behaviour* amounts to 5%, the *percentage of youngsters with criminal behaviour* amounts to 50%, the *percentage of adults with criminal behaviour* amounts to 60%, and the *percentage of retirees with criminal behaviour* amounts to 10%.

a. Make a SD simulation model of this description and verify it. Simulate a base run over a period of 50 years.
b. Now assume that due to *successful voluntary family planning measures*, the *birth flow* is 75% lower, that is, 25% times the sum of the *adults* times the *annual fertility rate of adults* of 3 percent per adult per year and the *youngsters* times the *annual fertility rate of youngster*s of 0.3 percent per youngster per year. Change the name of the run, and run it.
c. Compare the behaviour of the latter run to the behaviour of the base run. How much time do you think it would take before people would notice something is going on?
d. This simulation model only focuses on the first-order effects, that is, the reduction of individuals. The biggest gains, however, are likely to relate to higher-order effects such as the better socio-economic circumstances of those children that are born in families with fewer children that need to be taken care of. Make a highly aggregated CLD of the simulation model and extend it with higher-order effects. Use the CLD to explain the problem and possible solutions.

Exercise 4.11. Housing Stock Dynamics Example

After reading this SD book, you make a wise decision and start to work as a SD modelling consultant (you don't know it yet, but SD will become your passion soon). One of your first clients is the Dutch minister of Housing. Initially there are 5,000,000 *houses* in the Netherlands. After a period of dynamic equilibrium in the housing system, the minister foresees the need to expand the number of *houses* to 5,050,000 (an increase of 50,000 houses) and wants to understand the medium-term dynamics (about 8 years ahead) of this expansion of the housing stock after raising the *desired housing stock* with 50,000 units by month 20.

The minister wants you to make a SD simulation model of the current situation. As already said, the system is currently in dynamic equilibrium. The *houses* have *an average house lifetime* of 100 years, after which the houses are demolished. The *demolishing* flow equals the number of *houses* divided by the *average house lifetime*. All demolished houses are replaced by new houses. This takes place in two steps and takes some time. First of all, the number of houses that are being demolished per month enter a *planned houses* stock via a *planning* flow. The initial value of the *planned houses* stock equals the *planning* times a constant *average time from planning to building* of 3 months. The *planned houses* stock is emptied by a *building* flow that feeds the *houses under construction* stock. Model the *building* flow as the *planned houses* divided by the *average time from planning to building*. As initial value of the *houses under construction*, take the *building* flow times the *average time to build houses* of 6 months. The *completion* flow – which equals the stock of *houses under construction* divided by the *average time to build houses* – empties the *houses under construction* stock and feeds the *houses* stock. Check whether this model is in equilibrium.

If the model is indeed in equilibrium, then extend it as follows to test the expansion of the housing stock. Add a variable or constant *desired number of houses* and a variable called *housing gap*, which is equal to the *desired number of houses* minus the number of *houses*. The *housing gap* divided by the *average time to respond to the housing gap* of 8 months should now be added to the existing *planning* flow. Add the additional 50,000 desired houses with a STEP function.

a. Build the model following the instructions above, simulate it, and assess the dynamics of the system. Roughly how long does it take before the system is back in dynamic equilibrium?

b. Make graphs for the number of houses, the demolishing flow, the houses gap, the number of planned houses and the number of houses under construction.

c. Perform two validation tests, one for structure and one for behaviour. Assume that the goal of the model is a rough exploration. Describe the name of the test, how you executed it, what you observe, and your conclusion.

d. Draw a highly aggregated CLD of the system and use it to explain the link between the structure and the behaviour of the system. In other words: how could this behaviour be generated by the system's structure?

e. Contrary to the foreseen rise in the desired number of houses, things turn out differently: the housing market collapses due to an economic and financial crisis. Instead of raising the desired number of houses by 50000 houses, the minister decides to reduce the desired number of houses with 50000 houses (in the Netherlands, building permits are strongly regulated and social housing is huge). What happens?

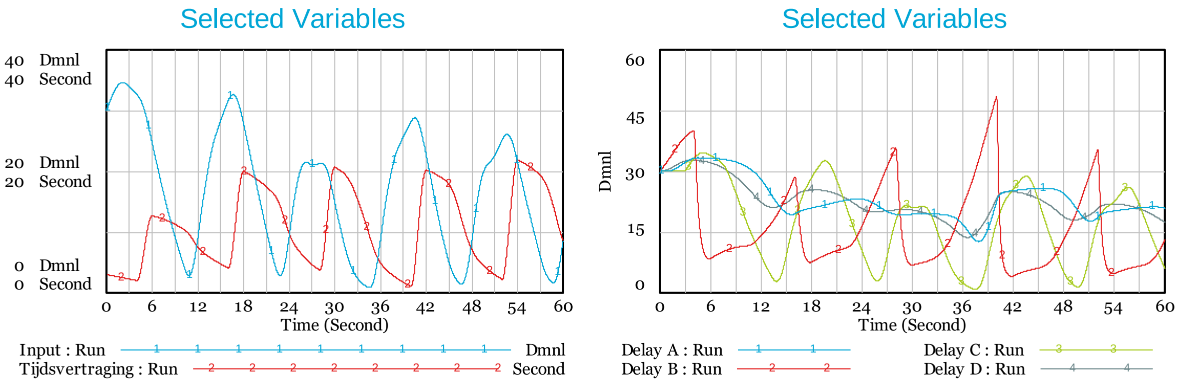
f. Twenty months later – that is, in month 40 – the same minister realises that the underlying desired outcome, which is housing for all (future) Dutch households, has not changed fundamentally. He suddenly decides to raise the level of houses to 5,050,000, which means raising it by 100,000 at the beginning of month 40. What are the consequences?

5. Model evaluation

5.1 Theory exercises

Exercise 5.1. Delays

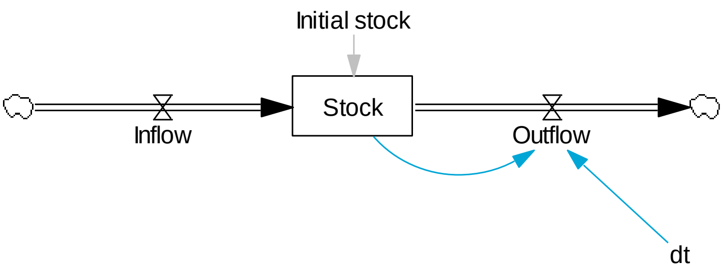
Consider the figure below with four outputs (right) of the same input and time delay (“Tijdsvertraging”, left).



a. What are the types of delays (material or information) and orders (1st, 3rd, or FIXED) from Delay A to Delay D?

Exercise 5.2. CLD to SFD (extra difficult)

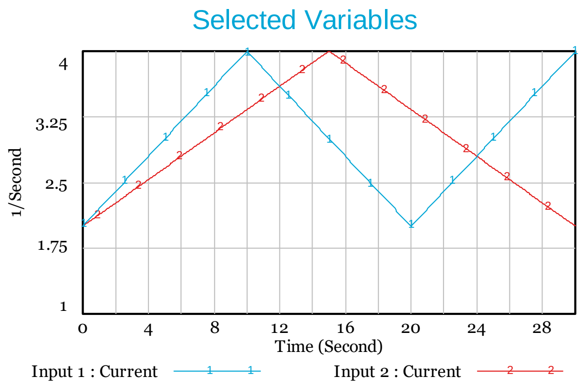
Consider the SFD below. The *Outflow* is defined as *DELAY1I (Stock / dt, dt, Stock / dt)*.

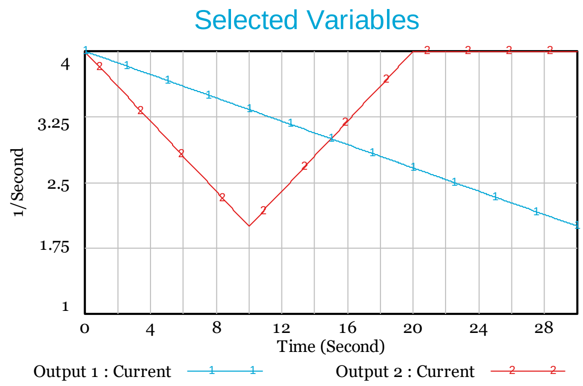


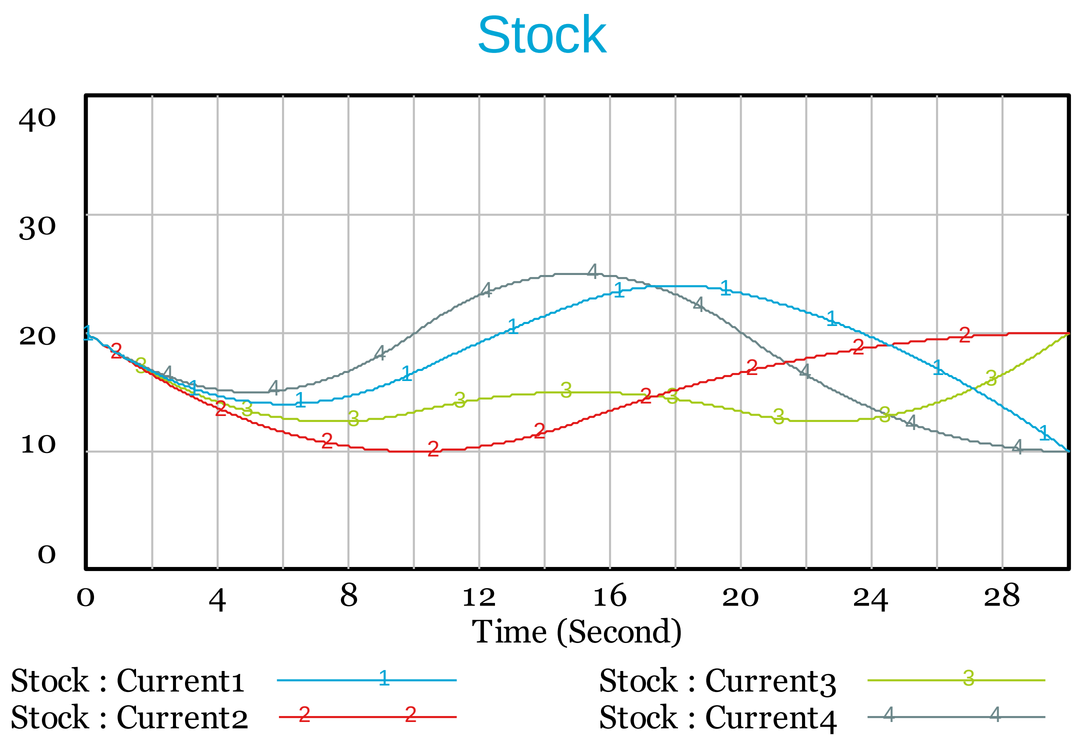
a. Draw the non-aggregated CLD of this system, using the information you have.

Exercise 5.3. Accumulation

Consider the following inputs (left) and outputs (right) and the behaviour of a stock with all possible combinations of these inputs and outputs (below).



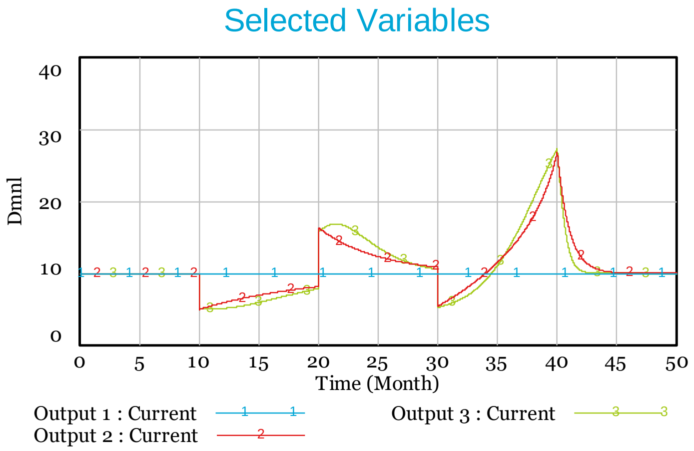




a. Indicate for each run what the corresponding input and output are.

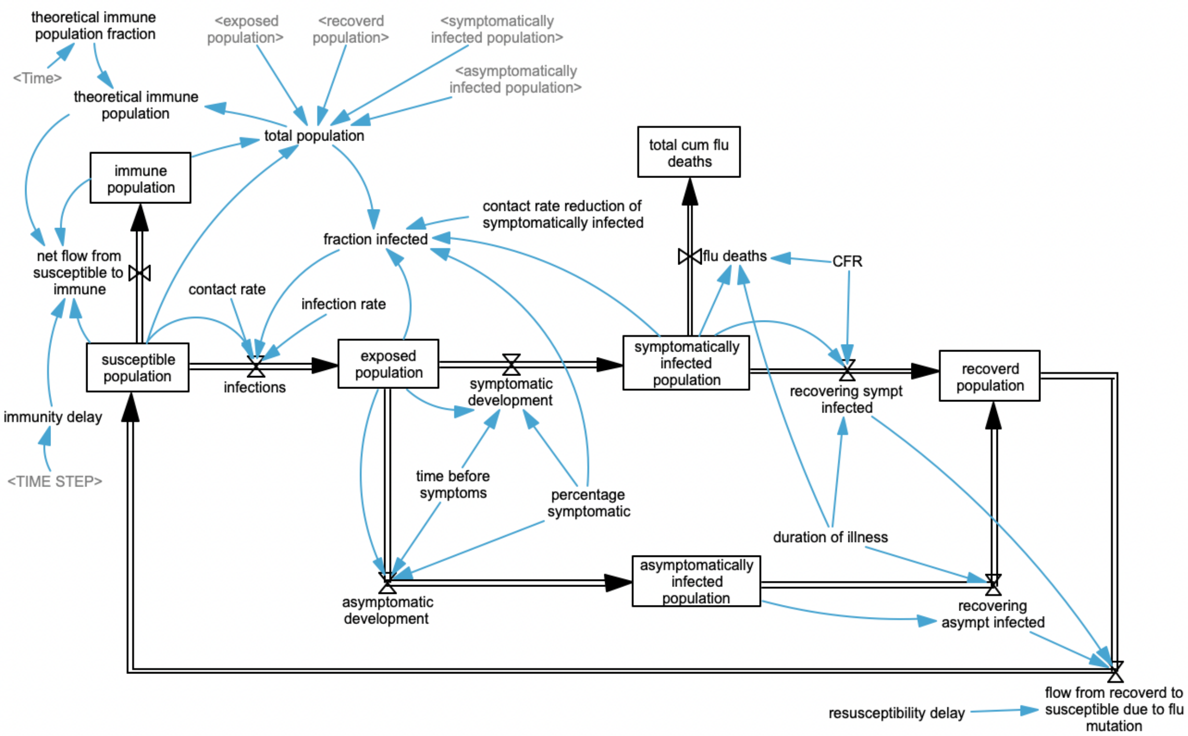
Exercise 5.4. Delays (extra difficult)

Consider the figure below with three different outputs of the same input and dynamic time delay.

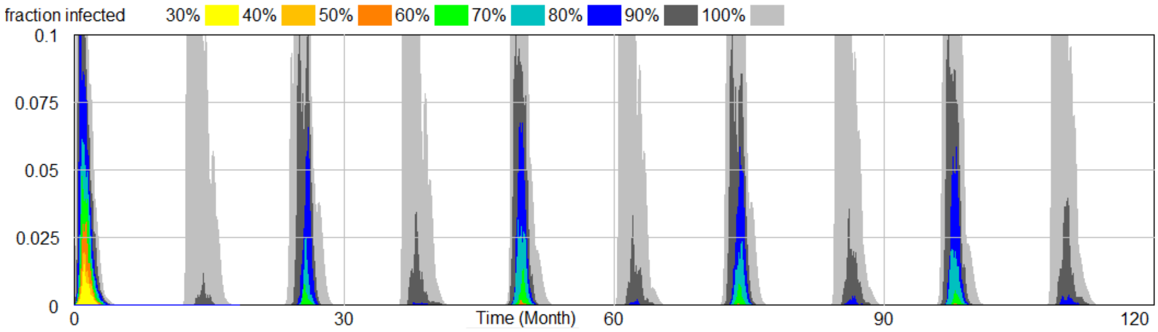


a. What are the delay types (material or information) of each output?
b. What are the delay orders (1st or 3rd) of Output 2 and Output 3?
c. What does the input of the delays look like? Create a graph of the input.

Exercise 5.5. Sensitivity analysis



Consider the seasonal flu model above. The sensitivity graph below was generated with this model, using large uncertainty intervals and Latin Hypercube sampling over 200 runs (see section 8.2.2).



a. What are your conclusions?

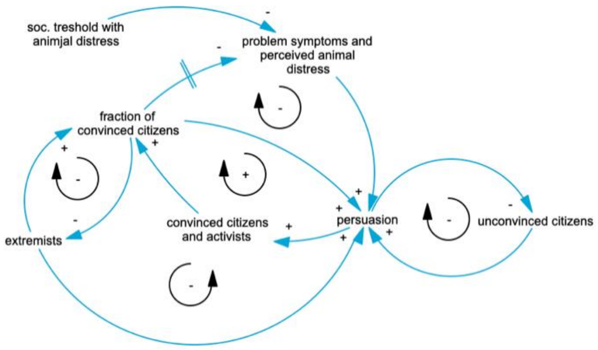
Exercise 5.6. Archetype

The real cause of the 2008 economic depression was – according to Nobel Prize winner Paul Krugman (2012, pp. 45-46) – systemic: '[I]f too many players in the economy find themselves in debt trouble at the same time, their collective efforts to get out of that trouble are self-defeating. If millions of troubled homeowners try to sell their houses to pay off their mortgages – or, for that matter, if their homes are seized by creditors, who then try to sell the foreclosed properties – the result is plunging home prices, which puts even more homeowners underwater and leads to even more forced sales. […] And if things get bad enough, the economy as a whole can suffer from deflation – falling prices across the board – which means that the purchasing power of the dollar rises, and hence that the real burden of debt rises even if the dollar value of debts is falling. […The latter] is the main explanation of the depression we're in right now.'

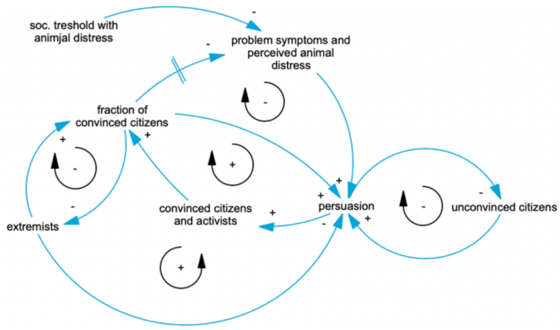
a. Which archetype corresponds best to this situation? Explain why in one sentence.

Exercise 5.7. Model behaviour

Consider the two CLD's below.



(a)



(b)

a. Which simulation models represented by CLD (a) and CLD (b) displayed above are most likely to lead to radicalisation and/or deradicalisation?

5.2 Modelling exercises

Exercise 5.8. COVID-19 Example

Introduction

The COVID-19 pandemic was arguably the biggest health crisis in over a hundred years, causing one of the biggest economic crises ever. It led to misery around the globe due to sickness, death and mental health issues related to the lockdowns imposed by governments in many countries.

Diseases like COVID-19 are often studied in so-called transmission models, which may be stochastic, like ABM models, or deterministic, as you will see here. Such models share a common SEIR structure (Susceptible, Exposed, Infectious, and Recovered), but are generally adapted to represent the specific functioning of the disease agent transmission in the case of interest.

Below is a description of a simulation model roughly mimicking the start of the COVID-19 pandemic in the Netherlands. This model is based on a standard deterministic transmission model, adapted for the specificities of SARS-CoV-2 transmission.

Simulate this SD model over a period of 60 weeks from the beginning of January 2020.

Static transmission model

We will first model the transmission model without any infections. The stock-flow structure of the transmission model is as follows. The *Susceptible population*, initially equal to 17 million persons, decreases by *Infection*, which increases the *Exposed population*, which is initially equal to 0 persons. For sake of simplicity, you are in this case allowed to not explicitly define these initial stock values, which are initially empty. Keep the *Infection* for now equal to 0. The *Exposed population* is decreased by *Incubation* to *Presymptomatic infectious*, initially 0, and by *Asymptomatic incubation* to *Asymptomatic infectious*, also initially 0. *Asymptomatic incubation* is defined as the *Share of asymptomatic infections*, which is believed to be 50%, times the *Total incubation*. The *Total incubation* is a third-order delay of *Infection* with the *Average SARS-CoV-2 incubation time* of 3 days divided by the number of *Days per week*. Choose a logical value for this last constant. *Incubation* is equal to 1 minus the *Share of asymptomatic infections*, times the *Total incubation*.

Presymptomatic infectious also increases by the *Influx of presymptomatic infectious people from abroad*. Keep this flow equal to 0 for now. *Presymptomatic infectious* develops into *Symptomatic infectious* (again, initially 0) by *Developing symptoms*. *Developing symptoms* is equal to a third-order delay of the sum of *Incubation* and the *Influx of presymptomatic infectious people from abroad*, delayed by the *Presymptomatic period* of 3 days divided by the *Days per week*. *Symptomatic infectious* can either go via *Recovery* to the *Recovered population*, or via *Dying of COVID-19* to the *Cumulative COVID-19 deaths*. Both these stocks are initially empty. *Dying of COVID-19* is equal to the product of the *Case fatality rate of COVID-19 patients* of 1% and the *Symptomatic infectious*, and is divided by the *Average length of symptomatic period* of 1 week. *Recovery* is thus to be defined as 1 minus the *Case fatality rate of COVID-19,* etc. The *Recovered population* also increases by the flow of *Recovery without symptoms*, which is equal to a third-order delay of *Asymptomatic incubation* with the *Average period of asymptomatic infections* of 6 days, divided by the number of *Days per week*. Finally, to complete this rather elaborate stock-flow structure, you have to include the effect of *Losing immunity*, which leads from the *Recovered population* back to the *Susceptible population*. *Losing immunity* is to be equated to the *Recovered population* divided by the *Average immunity period*, which we assume to be 52 weeks. The *Total population* should be equal to the sum of all stocks you have now, excluding the *Cumulative COVID-19 deaths*.

If you are building this model from scratch, check whether it simulates and save it.

Dynamic transmission model

To make the model dynamic, you first need to define the *Influx of presymptomatic infectious people from abroad*. This influx was created by people exposed to SARS-CoV-2 on holiday destinations in especially the Alps, and was equal to 0 in the beginning of the run time. It gradually increased to 70 in week 4.5, to 500 in week 7, 2000 in week 8, and 1000 in week 10. It then strongly reduced down to 80 in week 12, and hovered around 5 between week 14 and week 17, only to increase gradually due to more people going on holiday to 200 in week 26, 500 in week 34, and 200 in week 35. It reduced back to 5 in week 38, at which value it remained until week 52. After this, you may assume it gradually decreased to 0 in week 104.

Change the *Infection* to the product of the *Infectious population,* the number of *Contacts per person per week*, the *Infection rate per contact* (30%), and the *Chance of contact with an susceptible person*. The *Infectious population* is equal to the sum of the *Symptomatic infectious*, the *Presymptomatic infectious*, and the *Asymptomatic infectious*. This is special about SARS-CoV-2: most other viruses are assumed to only be infectious when symptomatic. The *Chance of contact with a susceptible person* is equal to the *Susceptible population* as fraction of the *Total population*. The *Contacts per person per week* is equal to the *Normal number of contacts per week* of 7.5 per week, times 1 minus the *Social distancing measures*. The *Social distancing measures* represent the effect of the interventions the Dutch government took to curb virus transmission by reducing the number of contacts people had.

The interventions were at 0% until week 8.9, when the advice was introduced to stop shaking hands at week 9. Assume this had a 10% effect until the end of that week. Then, the advice was given to work from home and stay 1.5 metres away from other people at all times. The maximum group size allowed was reduced in week 10. Assume this led to 40% reduction up to week 10.9. Next, schools and day care facilities were closed in week 11, which resulted in 70% reduction between that moment and the end of week 18 (week 18.9), when the measures were relaxed a bit by carefully reopening primary schools, making the reduction equal to 60% between week 19 and week 21.9. After that, in week 22, secondary schools, restaurants and bars were reopened, reducing the number to 55%. Gradually, this decreased due to further relaxation of the interventions, as well the population being less careful, to 50% in week 40.9. Increased infections led to renewed closing of restaurants and bars in week 41 till week 49.9 (leading to 55% reduction of contacts) and mandatory wearing of face masks, and finally a strict lockdown between week 50 and 55.9 (60% reduction). A curfew was enforced between week 56 and week 58 (70% reduction). After this, assume that measures were gradually relaxed till 0% contact reduction in week 104.

If you are building the model from scratch, the model should simulate again.

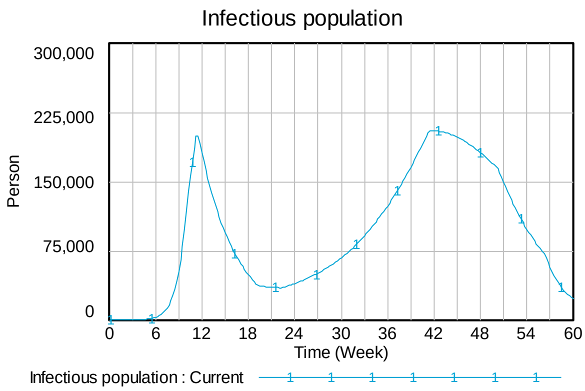
Testing and keeping track of the virus

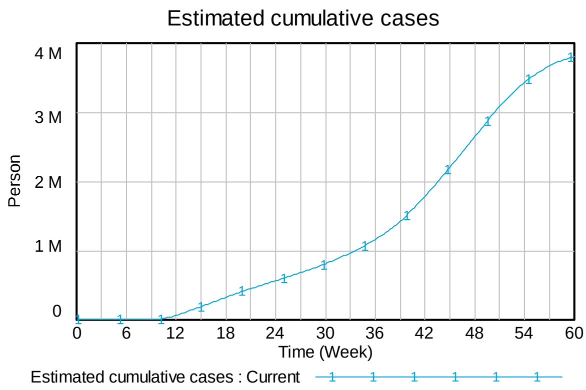
To keep track of the virus, the *Actual reproduction number* was continuously reported. This can be most easily calculated by the division of *Infection* by the *infectious population*. Make sure that this value is 0 when the *Infectious population* is equal to 0. Next, it is considered relevant to know the *Estimated cumulative cases*, which was of course initially 0, and increases by the *Cases tested* and *Additional case estimates*. The variable *Cases tested* is equal to the product of the *Symptomatic infectious* and the *Testing grade*, which increased from 0% per week in week 7, to 20% in week 10, 70% in week 40, and 80% in week 104. The *Additional case estimates* are equal to the sum of two information delays: first the *Asymptomatic infectious* divided by the *Measurement period* of 1 week delayed by the *Estimation period*, and second the *Symptomatic infectious* times 1 minus the *Testing grade*, also delayed by the *Estimation period*. The *Estimation period* decreased from 20 weeks in week 0 to 2 weeks in week 52 and to 1 week in week 104.

If you build the model from scratch, check whether your model simulates and has the right settings, and save it.

The model described above should have been modelled as precisely as possible. You can download several versions of this model from here. However, these SD models each contain three errors, despite the detailed description given. Please use the text and your common sense to correct the three errors.

a. Debug one of the incorrect COVID-19 models. The behaviour of the variables *Infectious population* and *Estimated cumulative cases* should be exactly like the two graphs below.





1. What are the correct model settings for this model?
2. What are the three errors in this model? List the variable names and corrected equations.

b. Draw a CLD of this model on a high aggregation level.

c. Perform a validation test of the structure of the model. Describe the name of the test, how you executed it, what you observe, and what your conclusion is.

d. Perform a validation test of the behaviour of the model. Describe name of the test, how you executed it, what you observe, and what the conclusion is.

e. What is unrealistic about this model? What could one do to improve the model, its usefulness, and the simulations it generates?

Exercise 5.9. Bank run

Example and explanation

Introduction

Many banks and financial institutions went bankrupt in 2009. The bankruptcy of a small Dutch bank, the Dirk Scheringa Bank or DSB Bank, is a very special case. The bankruptcy was actually caused by an orchestrated bank run by angry clients, following a call by Pieter Lakeman to empty their deposits.

Disclaimer: The model underlying this exercise is a very simplistic and highly aggregated model. It does not capture important processes and events such as a 'haircut', which we know – with hindsight – happened during the fall of DSB Bank. Hence, this is just an educational case: the model made in this case cannot be used to advise real banks. All values in this exercise are fictitious.

Modelling this Bank Run

Deposits being emptied are – from the point of view of a bank – *liquid deposits* and *loans lost*. These *liquid deposits* and *loans lost* drain the *liquid deposits and loans*, which initially amounted to €4,500,000,000.

Liquid assets lost are equal to the *liquid deposits and loans lost* because of the double accounting system. *Liquid assets lost* decrease the amount of *liquid assets*, which initially amounted to €1,150,000,000. Suppose that DSB had a *liquid asset liquid liability target* of 20%, which means that *fixed assets*, which initially amounted to €4,600,000,000, need to be liquidated and turned into *liquid assets* if less than 20% of *liquid deposits and loans* are covered by *liquid assets*. Suppose that the *liquidation time* is only 1 day, which means that there are enough parties willing to almost instantly buy these assets. However, in case of haste, there is a *liquidation premium* of 10% on emergency sales. In other words, only 90% of the fixed asset value is turned into *liquid assets* in emergency sales, and 10% of the fixed asset value is lost as *liquidation losses*. Keep in mind when you model the *liquidation* flow that the model you make is a crisis model and not a complete 'going-concern' banking model: there should be a net flow from *fixed assets* to *liquid assets*, but not the other way around. Apart from *liquid deposits and loans*, DSB also had *fixed deposits and loans* worth €1,000,000,000, which remained constant during the crisis because *fixed deposits and loans* cannot be emptied by depositors or lenders before their due date.

In a normal bank run, the number of *liquid deposits and loans lost* equals the *liquid fraction running away* times the *liquid deposits and loans* divided by the *withdrawal time*. However, two factors amplified the 'running away-effect' in the case of DSB. First, clients were angry because of unacceptable sales practices and the mediatized unwillingness of the bank to compensate the victims of these practices. Second, after having witnessed several bankruptcies in the banking world, people understood that a bank failure causes severe hindrance, since even if the clients of the bank get back their money back (in case of depositor guarantees), they have to wait months for this. Multiply the previous right-hand side of the equation, therefore, with the following two factors: $(1 + hindranceofbankfailures)$ and $(1 + anger)$. Suppose that the *hindrance of bank failures* amounts to 0.5.

Since the DSB bank run was triggered by a call for a bank run, you have to add an additional term to take this orchestrated action into account, for example: *orchestrated liquid fraction running away* times *liquid deposits and loans* divided by a *withdrawal time* of 1 day. Note that the maximum number of *liquid deposits and loans lost* equals the number of *liquid deposits and loans* divided by the *withdrawal time*.

Suppose that the *liquid fraction running away* amounts to 0% if the *perceived likelihood of a bank failure* is 0%, that it amounts to 0% if the *perceived likelihood of a bank failure* is 25%, to 1% if the *perceived likelihood of a bank failure* is 50%, to 10% if the *perceived likelihood of a bank failure* is 75%, and to 50% if the *perceived likelihood of a bank failure* is 100%. Model the *perceived likelihood of a bank failure* as $(100\% - credibilityofthedenials)$ times the maximum of either the *perceived likelihood of a liquidity failure* or the *perceived likelihood of a solvency failure*.

Suppose that the *perceived likelihood of a liquidity failure* amounts to 100% if the *liquid asset liquid liability ratio* equals -1, to 100% if the ratio equals 0, to 80% if the ratio equals 0.1, to 40% if the ratio equals 0.2, to 10% if the ratio equals 0.3, to 1% if the ratio equals 0.4, 0% if the ratio equals 0.5, and 0% if the ratio equals 1.

Suppose that the *perceived likelihood of a solvency failure* amounts to 100% if the *total asset total liability ratio* equals 0, to 100% if the ratio equals 0.8, to 90% if the ratio equals 0.9, to 50% if the ratio equals 1, to 10% if the ratio equals 1.1, to 0% if the ratio equals 1.2, and to 0% if the ratio equals 2.

The *liquid asset liquid liability ratio* is of course equal to the amount of *liquid assets* over the amount of *liquid deposits and loans*. And the *total asset total liability ratio* is equal to the sum of the *fixed assets* and the *liquid assets* over the sum of the *liquid deposits and loans* and the *fixed deposits and loans*.

Suppose the central bank issues a *bank failure declaration*, forcing a bank into bankruptcy, if the *liquid asset liquid liability ratio* falls below 0.05 or the *total asset total liability ratio* falls below 0.9.

a. Make a SD simulation model of this issue. Verify the model.

b. Simulate the model, firstly, over a time horizon of about 60 days without any angry customers calling for an 'organized' bank run. In other words, set *anger* equal to 0, *orchestrated liquid fraction running away* equal to 0%, and the *credibility of the denials* equal to 90%. Plot the *liquid deposits and loans lost* and the *perceived likelihood of a bank failure*.

c. Now, adapt the model to simulate a bank run following Pieter Lakeman's call for an orchestrated bank run. Suppose for example that the *orchestrated liquid fraction running away* jumps to 5% on day 2 and falls to 0% on day 4, and that the *credibility of the denials* falls from 90% to 10% from day 2 on. Suppose also that on top of these changes the variable *anger* amounts to 0.5 and the *liquidation premium* to 25%.

    1. Simulate the model over a time horizon of 60 days. Plot the *liquid deposits and loans lost* and the *perceived likelihood of a bank failure*.
    2. Explain the behaviours obtained, especially if you obtain strange behaviours.
    3. Briefly describe if the bank could have done anything do to prevent this bank run, and if so, what (do not model it here).

d. Simulate a bad case scenario, again over a time horizon of 60 days, in which *anger* is 1, *hindrance of bank failures* is 1, and the *liquidation premium* is 25%.

    1. Plot the *liquid deposits and loans lost* and the *perceived likelihood of a bank failure*.
    2. Briefly describe the differences with the previous behaviour.
    3. Briefly describe whether if the bank could have done anything do to prevent this bank run, and if so, what (do not model it here).

e. Validate the model. Use two different validation tests. List the tests used, how you performed them, and describe the conclusions of these tests.

f. Draw an aggregated CLD of the system to help you communicate the main feedback effects responsible for the bank run after the call.

g. Explain the link between structure and behaviour (e.g., for the 'bad case' scenario).

h. Add a simple closed-loop or adaptive policy that prevents the bank from collapsing. Describe the policy briefly. Test the policy in the 'bad case' scenario and plot the resulting dynamics together with the original dynamics.

i. What do we call variables like *hindrance of bank failures* and *anger*?

Exercise 5.10. Activism, Extremism and Terrorism

Example and explanation[2]

Introduction

You are asked to make a more detailed model focused on phenomenon-based activism and extremism, for example animal rights activism and extremism.

Citizens, Activists, and Extremists

Just as in the more generic model, you can simplify the population sub-model by assuming that the population remains constant (no migration, no births and no deaths). Suppose that there were 16 million citizens in the Netherlands in 1980, of which about 3 million were *citizens that cannot be convinced* that animals should have any rights; 12900000 were initially *unconvinced citizens*, and about 100000 were initially *convinced citizens*, who were already sympathetic towards animal rights. Initially, there were no *activists* and *extremists*. *Unconvinced citizens* become *convinced citizens* through *persuasion*. *Convinced citizens* could possibly become – still law-abiding – *activists* through *activation*. After some time, some *activists* start to operate outside of the legal framework and thus become *extremists*. The latter process may be called *extremisation*.

The *persuasion* variable could be modelled as the product of the *contact rate of convinced with unconvinced citizens*, the *fraction of all convinced citizens*, the *unconvinced citizens* that are not (yet) convinced but possibly could be, the *persuasion rate* of 1%, and the *reinforced visibility of animal distress* divided by an *average transition time* of 10 years. Note that the *persuasion* flow cannot be greater than the number of *unconvinced citizens* divided by an *average transition time*. Set the *contact rate of convinced with unconvinced citizens* equal to the *frustration of all convinced citizens* times the *normal contact rate of convinced* of 1000 contacts per convinced citizen per year.

Simplify the *activation* and *extremisation* flows. Assume the *activation* flow is equal to the difference between the *potential number of activists* and the number of *activists*, divided by the *average transition time*, and that the *extremisation* flow is equal to the difference between the *potential number of extremists* and the number of *extremists*, divided by the *average transition time*.

The *potential number of activists* is determined by the product of the *potential fraction of activists* of 5%, the number of *convinced citizens*, and the *frustration of all convinced citizens*. Model the *potential number of extremists* as the product of the

[2] Be careful: the model in the link does not completely correspond with the description below.

potential fraction of extremists of 5%, the number of *activists*, the *frustration of all convinced citizens*, and the *frustration through marginalization*.

Human Distress

The *frustration of all convinced citizens* – always between 0% and 100% – can be modelled as the product of the *perceived animal distress* and the sum of the *frustration through marginalization* and the *frustration through inertia*. The *frustration through marginalization* is a function of the *fraction of all convinced citizens* connecting the following couples: (0, 1), (0.025, 0.50), (0.10, 0.20), (0.25, 0.04), (0.5, 0), (1, 0). The *fraction of all convinced citizens* is of course equal to the sum of *convinced citizens*, *activists*, and *extremists*, divided by the total population.

Model the *frustration through inertia* as the difference between the *maximum attainable rate of decrease through societal change* of 5% and the *rate of decrease through societal change*, divided by the *maximum attainable rate of decrease through societal change*, but then delayed by exactly one year (because statistics are always delayed and newspapers report statistics after publication). This *rate of decrease through societal change* is equal to the *fraction of all convinced citizens* times the *maximum attainable rate of decrease through societal change*.

Animal Distress and Visibility

Perceived animal distress could be modelled as the *real animal distress* divided by the *societal acceptance threshold with regard to animal distress*, but then smoothed (3^{rd} order over a year). Suppose that the *societal acceptance threshold with regard to animal distress* changes over time from 100 in 1980, to 80 in 2000, to 40 in 2020, to 20 in 2040, to 12 in 2060, and to 10 in 2080.

Real animal distress – in 1980 equal to the reference value 100 – increases or decreases through the *net increase of animal distress*, which can be modelled as the product of a*nimal distress* and the difference between the *exogenous rate of increase of animal distress* and the *rate of decrease through societal change*. The *exogenous rate of increase of animal distress* amounts to -0.1%; in other words, animal distress decreases exogenously.

Model the *visible animal distress* as the difference between *real animal distress* and the *societal acceptance threshold with regard to animal distress*, divided by the *societal acceptance threshold with regard to animal distress*. The *visible animal distress* should always be non-negative. The *visible animal distress* multiplied by the *radical action level* results in the *reinforced visibility of animal distress*. The *radical action level* is equal to the number of *extremists* times the *frustration of all convinced citizens* times the *fraction of all convinced citizens*.

a. Make a model based on the description above. Verify it. Then simulate the model over a time horizon of 100 years, starting in 1980.

b. Make a graph of the *convinced citizens*, *activists*, and *extremists*, and a graph of the *persuasion* rate and the *frustration of all convinced citizens*.
c. Validate the model: name, describe and perform two different validation tests (except sensitivity testing – see following question) for a model like this. What conclusion can you draw from this?
d. Perform a multi-variate sensitivity analysis. Briefly describe the tests performed, what you observe, and your conclusions.
e. Make different consistent and interesting (i.e., behaviourally different) scenarios, starting from your conclusions in the previous question. Simulate these scenarios and draw graphs for two scenarios of *convinced citizens*, *activists*, and *extremists*, and the *persuasion* rate and the *frustration of all convinced citizens*.
f. Make a highly aggregated CLD of this model that could be used to explain the link between the structure and behaviour of one of the scenarios. Use this CLD to explain that behaviour.
g. Formulate policy advice – based on this exercise – with regard to animal rights activism.
h. This model is only a simple starting model, and the analysis is explorative at most. Provide advice with regard to future extensions and refinements of the model.

Exercise 5.11. Higher Education Stimuli

[Example and explanation]

Introduction

A mass demonstration was organized on 21 January 2011 in the Netherlands in order to protest against proposed legislation to fine students taking more time for their degrees than the standard set by the government, and to also fine universities with ‘slow’ students. Students at the Faculty of Technology, Policy and Management at TU Delft – who are bright but also slow according to the definition of ‘slow students’ – are asked to model the potential consequences of such national policies for the faculty, based on the description below.

BSc Students

First, model the inflow of BSc students. There is an *annual BSc inflow* at the *BSc inflow moment*. Suppose for reasons of simplicity that there is one inflow moment per year. Use a PULSE TRAIN(start, width, time between, end) with a *width* equal to the *time step*. Model the normal *annual BSc inflow* as the *normal evolution of the new BSc inflow* divided by the *time step* times the *BSc inflow moment*. Suppose that the *normal evolution of the new BSc inflow* gradually increased from 20 new BSc students when the faculty was founded in 1990, to 90 new BSc students in 1995, to 120 new BSc students in 2000, to 130 new BSc students in 2008, to 200 new BSc students in 2010, and to 250 BSc students in 2014, and that it is believed to stabilize at 250 from the year 2030 onwards. The *real inflow of BSc students* is the product of the *annual BSc inflow* and the *quality*: the lower the quality of the education, the lower the real inflow will be. For now, set the *quality* equal to 100%.

The *real inflow of BSc students* is added to the group of *BSc students*. The group of *BSc students* decreases through the *outflow BSc students* if students obtain their BSc or when they quit before graduating as *BSc quitters*. Model the outflow of *BSc quitters* simplistically (but not entirely correctly) as the *fraction of BSc quitters* times the *BSc outflow after the fixed and additional study time*. Suppose that 30% of the students quits during the first year, 10% in the second year, and 5% in the third year. The *fraction of BSc quitters* – always between 0 and 1 – is, therefore, the sum of these fractions divided by the *quality:* the lower the quality of the education, the more quitters. Those who do not drop out, obtain their BSc diploma eventually. The *outflow of BSc students*, therefore, equals the *BSc outflow after fixed and additional study time* multiplied by the complement of the *fraction of BSc quitters*. Model the *BSc outflow after fixed and additional study time* as the first-order delay of the *BSc outflow without additional study time* with a total delay time equal to the product of the *minimum BSc study time* of 3 years and the *additional annual BSc study time* of (on average) 50% divided by the *quality.* And model the variable *BSc outflow without additional study time* as the delay of the *inflow of BSc students* with a fixed *minimum BSc study time* of exactly 3 years.

a. Make a SD model based on the description above.
b. Make a complete CLD and a strongly aggregated CLD of this partial simulation model.

MSc Students

Now, model the throughput of MSc students. Almost the same applies to MSc students as to BSc students, but the following details are different. The *real inflow of MSc students* equals the *quality* of the education times the sum of the *annual MSc inflow of new MSc students* (who do not flow semi-automatically from BSc to MSc studies) and the product of the *outflow of BSc students* and the *fraction of BSc student*s that flow semi-automatically from the BSc to the MSc program. The *evolution of the new MSc inflow* was 0 students per year until 2007, then started with 2 students per year in 2008, and rose to 5 students per year in 2010, to 15 students per year in 2015, and 20 in 2020, after it remained constant. The *fraction of students* (that flow semi-automatically) *from the BSc to the MSc* program was about 100% before the year 2008; suppose that it fell to 80% of the students in 2008 and afterwards. The *minimum MSc study time* is equal to 2 years. The *fraction of MSc quitters* – always between 0 and 1 – is lower than for BSc students: 10% in the first year and 10% in the second year. In summary: the structure of the MSc students sub-model is the same as the BSc students sub-model: a handful of new MSc students is absorbed by a larger, but decreasing group of students flowing semi-automatically from the BSc to the MSc program.

c. Extend the SD model using the description above.

The Faculty

The *quality* of the education is a function of the *professor hours per student*: if the number of *professor hours per student* is 0 hours per student per year, the *quality* is 10%, if it is 50 hours per student per year then the *quality* is 60%, if it is 100 hours per student per year then the *quality* is 90%, if it is 150 or more hours per student per year then the *quality* is 100%.

Model the *professor hours per student* as a 3rd order delay with one year of the product of 1000 hours per professor and the number of *professors* divided by the *total number of students*. Make sure in the previous formula that the denominator cannot become 0.

Model the number of *professors*, starting initially at 5 in 1990, and the increase and decrease of the number of professors in a simplistic way. Suppose that *net hiring of professors* equals the difference between the *maximum number of professors* and the number of *professors*, divided by the average net *hiring time*. The *hiring time* (and firing time) for professors is rather long – they are hired on average 2 years after the moment a new professor is needed. The *maximum number of professors* then equals the *amount of money available for education* divided by the *average professor salary* of €100000 per professor per year.

The *amount of money available for education* – initially 0 – increases through the *inflow of money available for education* and decreases through the *outflow of money available for education*. Without a fine for slow students, the *outflow of money available for education* approximately amounts to the number of *professor*s times the *average professor salary*.

The *fraction of slow students* seems to be – at least partly – a function of the *quality* of the education: if the *quality* is 0% then the *fraction of slow students* is 90%, if the *quality* is 25% then the fraction is equal to 85%, if the *quality* is 50% then the fraction is equal to 66%, if the *quality* is 75% then the fraction is equal to 40%, and if the *quality* is equal to 100% then the fraction is equal to 25%.

The *inflow of money available for education* equals the *subsidy per new BSc student* times the *inflow of BSc students*, plus *the subsidy per BSc graduate* times the *outflow BSc students*, plus the *subsidy per new MSc student* times the *inflow of MSc students*, plus the *subsidy per MSc graduate* times the *outflow of MSc students*, plus the *annual lump sum and other subsidies*. The *subsidy per new BSc student* amounts to €15000, the *subsidy per BSc graduate* to €5000, the *subsidy per new MSc student* €5000, and the *subsidy per MSc graduate* €5000. Suppose that the *annual lump sum and other subsidies* for educational purposes amount to an additional €1 million per year.

d. Extend the SD model based on the description above.

e. Simulate the model without fines for slow students from the year 1990 until the year 2030. Make graphs of following variables: *BSc students* and *MSc students*, *outflow of BSc students* and *outflow of MSc students*, *professors*, and the amount of *money available for education*. Is the faculty healthy without the proposed system of fines?

And now with fines for slow students…

However, what does the proposed system of fines for slow students mean for the faculty? Model the system of fines as follows. With the fine system, the *outflow of money available for education* changes to the *fraction of slow students* times the *total number of students* times the *fine per slow student* plus the number of *professors* times the *average professor salary*. Expect (at least) a one year delay before implementation due to opposition and demonstrations: you can assume that fines for slow students were introduced in 2012 and that the *fine per slow student* amounts to €3000 per year. This is of course only part of the picture: increased tuition fees to be paid by slow students are not taken into account here.

f. Extend the SD model with the description provided above.

g. Simulate the model with the system of fines. Make graphs of following variables: *BSc students* and *MSc students*, *outflow of BSc students* and *outflow of MSc students*, *professors*, and the *amount of money available for education*. Is the faculty financially healthy with the proposed system of fines?

h. What would be the effect if the number of new students increased to 300 new BSc students per year and to 60 new MSc students per year in 2020?

i. Your model is used by your university and national student council to fight these plans. The government is furious: they claim your model is wrong, because the *outflow of money available for education* still needs to be divided by a factor 2.5 (the average number of study years for BSc and MSc students). Is that correct? Do the conclusions change?

j. After the previous proposed correction has been made, the government now argues that the LOOKUP function of the *fraction of slow students* needs to be adapted too, since the government assumes, after all, that students will study faster in the new system. Change the lookup, discuss the new function and the consequences of this change for the faculty.

k. It should be clear that this model requires further adaptation. How could you make it more realistic? Describe what and how should be added; model these, and describe the possible changes in results.

From Fining to Lending

When implementing the fine system, the Dutch government first turned the slow students fine forced upon universities into a lump sum cut in their subsidies. Then, during the first year of imposing an individual fine on slow students, the system was abolished. This resulted in big losses, both financial and human: financial systems had already been changed, students had dropped out, et cetera. The fine system was then turned into a social lending system. Nowadays, Dutch students can borrow money to finance their studies.

l. If you feel like it, model the new system. What could be you conclusion?

m. An even bigger challenge would be to model and simulate the political process. From this process, much can be learned about how not to change a system but nevertheless stay in power.

Exercise 5.12. Financial Turmoil on the Housing Market

[Example and explanation]

Introduction

The Dutch housing market has been in crisis for a while and will most likely remain in crisis for several years to come: average real estate prices have increased enormously; the largest mortgage lenders made it more difficult to get a mortgage, and new housing construction is only a fraction of what is needed. Policymakers, therefore, gain insight into how the existing shortage on the housing market may evolve in the medium to long term. Suppose that the Ministry of Housing asks you to develop a SD model that would allow them to foresee the evolution of the Dutch housing market and to assess policies related to it.

Iteration I

Assume for the sake of simplicity that the Dutch housing market only consists of houses for sale (no apartments, no rental market, and no social housing market).

Houses are either *new* (younger than 15 years old) or *old* (15 years or older). The supply of *new houses*, equal to 1500000 in the year 1985, increases through *completion of brand new houses* and decreases through *aging of houses*, after which they are added to the *old houses*. The supply of *old houses*, initially 3665000 houses in 1985, decreases through *demolishing of old houses*.

The *completion of brand new houses* could be approximated by dividing the *houses in planning and under construction* by the *planning and construction time*. The number of *houses in planning and under construction* increases only through the *inflow into planning and construction* and decreases only through the *completion of brand new houses*. Suppose that the initial number of *houses in planning and under construction* in 1985 was 175000. Suppose that the *planning and construction time* is a function of the number of *houses in planning and under construction* divided by the *initial amount of houses in planning and under construction*: the *planning and construction time* equals 1 year if this ratio is equal to 1, equals 1.5 years if the ratio is equal to 2, equals 2.5 years if the ratio is equal to 5, equals 3.25 years if the ratio is equal to 9, equals 4.5 years if the ratio is equal to 20, and equals 0.75 years if the ratio is close to 0.

The *inflow into planning and construction* could be modelled as the *housing gap* multiplied by the *profitability multiplier* and divided by the *delayed direct effect of uncertainty*. Suppose in this first iteration that the *profitability multiplier* is 1. Model the *delayed direct effect of uncertainty* as a 3rd order delay of *uncertainty* with a delay time of 1 year. Assume the *uncertainty* on the housing market was normal (i.e., 100%) from 1985 until mid-2007, after which uncertainty suddenly doubled. The uncertainty remained this high until the end of 2013, and decreased linearly from double to normal between the end of 2013 and the beginning of 2015, and remained normal after this.

The non-negative *housing gap* is equal to the *estimated number of households* times the number of *houses per household* minus the *expected total housing supply*. The *estimated number of households* amounted to 5,430,000 in 1985, to 5978000 in 1990, to 6798000 in 2000, to 7397000 in 2010, to 7470000 in 2011, and is assumed to amount to 8500000 in 2050, and to 9000000 in 2085. Assume that households do not have more than one house: the number of *houses per household* is 1. The *expected total housing supply* is the sum of *new houses*, *houses in planning and under construction*, and *old houses*, minus the houses expected to be demolished over the course of the year.

The *aging of houses* follows the *completion of brand new houses* with a delay time equal to the *life expectancy as new houses* of exactly 15 years. Model the *demolishing of old houses* as the number of *old houses* over the *average life expectancy of old houses* of about 60 years multiplied with a *demolishing multiplier of old houses*. Suppose the latter multiplier could be modelled as the 3rd order delay of 1 divided by the *housing scarcity ratio* with a delay time of 1 year. The *housing scarcity ratio* is directly proportional to the *estimated number of households* and inversely proportional to the *expected total housing supply*.

a. Model the description above.
b. Simulate the behaviour from the year 1985 until the year 2085.
c. Draw an extremely aggregated CLD which could be used to explain the general dynamics of the *housing gap*. Explain the general dynamics of the housing gap using, and referring to, this extremely aggregated CLD.

Iteration II

Suppose that the *profitability multiplier* is a function of the *profitability of construction of new housing* in such a way that this multiplier equals 0 if the *profitability of construction of new housing* is equal to -100%, that it equals 0.01 if the profitability is equal to -50%, that it equals 0.02 if the profitability is equal to -20%, that it equals 0.2 if the profitability is equal to -10%, that it equals 0.8 if the profitability is equal to 0, that it equals 1 if the profitability is equal to 10%, that it equals 1.1 if the profitability is equal to 20%, that it equals 1.2 if the profitability is equal to 50%, and that it equals 1.25 if the profitability is equal to 100%.

The *profitability of construction of new housing* could be formulated simplistically as:

$$\frac{(1 + acceptable\%additionalcostforliving \in anewhouse) * averagehousep}{constructioncostnewhouse} \quad \text{Eq. 5.2}$$

with an *acceptable % additional cost for living in a new house* of 5%. The *average construction cost of a new house* equals the *initial average construction cost of a new house* of €95000 per house times the cumulative inflation since 1985. *The cumulative inflation since 1985* could be calculated as the integral of:

$$inflationrate * cumulativeinflationsince1985 * MAX\left(\left(1-(uncertainty-1)\right), 0\right), \quad \text{Eq. 5.3}$$

with an initial value equal to 1. Assume for the sake of simplicity that the *inflation rate* is standard at 2% per year. The *average house price* corresponds – given the simplification that every household has no more than 1 house – to the delayed product of *the average spending limit for buying one house per household* and *the housing scarcity ratio*, with an average delay of one year. The *average spending limit for buying one house per household* equals the *average salary per household* times (1 + *salary loan multiplier*). Suppose that the *average salary per household* is the product of the *initial average salary per working person* of €27,000 per year in 1985, the *cumulative inflation since 1985*, and the *expected work force* divided by the *estimated number of households*. Add the following time series: suppose that the *expected work force* amounts to 5.75 million in 1985, to 7.5 million in 2012, to 8 million in 2020, to 7.6 million in 2040, to 7.3 million in 2050, and to 6 million in 2085. The *salary loan multiplier* used to calculate the average mortgage lending capacity of an average household is then:

$$\frac{normalsalaryloanmultiplier * (1 - loanrisk)}{delayeddirecteffectofuncertainty} \qquad \text{Eq. 5.4}$$

The *normal salary loan multiplier* increased linearly between 1985 and the end of 2011 from 3 to 6, but decreased to 4 between 2011 and 2013 because of stricter rules for banks and mortgage transactions, and the gradual decline of the mortgage interest relief. It is expected to slowly fall back to 3.5 by 2050 and stay at that level afterwards. Suppose finally that the loan risk (i.e., the risk of non-repayment of loans), could be approximated by a 3rd order delay of MAX(0; *uncertainty* − 1)/6 with an average delay time of 2 years.

d. Model the description above. Verify and simulate the model. Compare the dynamics of the *average house price* and the *housing gap* of the first and the second iteration model.

e. Validate the second iteration model: name two good validation test for this phase in the modelling process, execute them, and describe the results.

f. Some inputs, such as exogenous future evolutions (time series) and endogenous relations are rather uncertain. Test the sensitivity of the most important indicators (*average house price* and *housing gap*) for changes in at least one uncertain parameter and one uncertain time series or endogenous relation. Explain the results briefly: what happens and why, how it happens, and what the results are?

g. Simulate – on top of the base case scenario – two very different scenarios with respect to the *average house price* and/or the *housing gap*. What is the narrative of the three scenarios? Plot their dynamics for the two key performance indicators.

h. What is undesirable about these plausible futures? Design a policy to turn undesirable dynamics into desirable dynamics. Describe this briefly, test it in your model, and compare the undesirable and desirable dynamics that result from it.

i. Test this policy under uncertainty: simulate the model, briefly describe the results, and draw your conclusions. Next, write a short model-based recommendation concerning the real estate market: what is your advice to the government, to current homeowners, and to those looking to buy a house?

Theory

6. Problem Formulation and Conceptualisation

This chapter focuses on conceptualisation as used in System Dynamics. The goal of the conceptualisation phase of the modelling process is to capture the feedback structure that can offer an endogenous explanation of the problem (Sterman, 2000). The conceptualisation phase results in one or several graphical conceptual models. These conceptual models are the basis for a quantitative specification of a system, but the models themselves can also be used for a qualitative analysis of system behaviour. The result of this qualitative analysis usually includes a theory of how the structure gives rise to the problem. Please note that this book only describes models that are suitable for systems that can be regarded as continuous.

6.1 Problem Formulation

The definition of the model objective has significant implications for the final model. An SD model is developed to understand the combination of forces that are causing the problem. To create a useful model, there must be an underlying problem that ensures a demand for improvement in the understanding of a system. After determining the problem field on which the model is focused, the modeller must collect information and define the function of the model in detail. This not only involves the collection of data points, but also information obtained from people who are familiar with the system involved.

The modeller must also consider the people for whom the model is intended. A model relating to the causes of acid rain for a high school class will differ considerably from a model for the Ministry of Housing. If the structure and behaviour of a model are not understood by the people for whom the model is intended or if the model fails to answer the required questions, the model will be useless. This might seem obvious, but in practice it has been shown that it is very difficult to find out precisely what people are interested in. Only after the modeller has made a prototype of the model, the questions people are really interested in suddenly surface, because before that they could not imagine exactly what the model would do or they had not considered the situation thoroughly enough.

Consequently, it is important to pay sufficient attention to the problem articulation and conceptualisation phase. Without a clearly described modelling objective, it is very difficult to determine which elements and aspects of the system are important. One has to ask oneself whether it is worth the effort to develop the model according to this description. An objective such as: “this model of the environment is intended to provide an understanding of the situation”, is very likely a waste of time. Such a description is too abstract and vague and, as a result, it is not clear where one must start and what definition is required. Nor should the modeller attempt to develop a model that applies to all situations and that must answer both general and specific questions, as the model would become too detailed and complex to be used for a practical analysis.

The model objective usually falls into one or more of the following categories:

- Understanding and exploring plausible system behaviour;
- Finding measures for improving system behaviour;
- Hypothesis testing (Richardson & Pugh, 1981; Sterman, 2000);
- Specifying mental models and serving as a communication tool.

Example: Drug-Crime System

A problem in combating drug crime is that if the police carries out more drug busts, this results in the undesired side effect of a rise in crime figures. In view of the broad problem field of the heroin-crime system, a model of drug busts and crime might be built for many different reasons. For example, if the modeller wants to contribute to a newspaper article on this problem, the objective of the model is studying if and why an increase in the number of drug busts in the short-term results in an increase of the number of drug-related crimes. In this case, the model is intended for the average person, and this target audience is interested in a simple model with aggregated components (i.e., on a low level of detail). If a model were to be built for a panel of experts in drug trafficking, aiming to reduce the number of drug-related crimes, it would need to be different.

6.2 Motivating the Use of SD

In your communication about your modelling research, you need to make insightful why you use, or want to use, SD. The stocks, flows, and feedbacks in SD models provide the reasons why an SD approach might be appropriate. The stocks allow the accumulation of important states in the system, like the size of different population cohorts, the number of houses, or the capacity for resource extraction. Stocks and flows together cause delays in the system. For example, it takes time for people to age, houses take time to be completed, and supply and demand of materials show a delayed reaction to price dynamics. Finally, feedback is often considered to be the most important reason for applying SD. Feedback may exist between people in fertile age, who cause births of new children, who age and in time become people childbearing age (a reinforcing feedback). Alternatively, an increase in houses causes housing prices to decrease, which causes fewer new housing projects to be developed, leading to fewer new houses (balancing feedback). Similar feedback can be recognised in resource supply systems.

The complexity caused by accumulation, delay, and feedback renders mental simulation of the problems formulated in the first phase of the research impossible

(Sterman, 1994). This is why it is justified to use simulation models, which are time-consuming and difficult (i.e., costly) to develop. For these problems, SD models can function almost as psychedelics, as they allow you to think – in a structured manner and based on available knowledge – beyond your mental simulation and come to new insights.

6.3 Conceptualisation

The following steps are performed during the conceptualisation phase of an SD project:

- Determining model boundaries and identifying the important variables;
- Creating diagrams of the major mechanisms, including feedback loops;
- Formulating a dynamic hypothesis, describing the expected behaviour of the system linked to the system structure (Sterman, 2000).

Some authors (e.g., Randers, 1980) consider the determination of the model's objective and function to be part of conceptualisation, while most others (Auping, 2018; Forrester, 1994; Keating, 1998; Richardson & Pugh, 1981; Sterman, 2000) consider it to be a step prior to conceptualisation.

6.4 Diagrammatic Conventions

Different types of diagrams are used in SD, both for model conceptualisation and model communication, and even for purely qualitative SD. Most often used are the Causal Loop Diagram (CLD), Stock-Flow Diagram (SFD), Sub-System Diagram (SSD), Bull's Eye Diagram, Influence Diagram (ID), and Archetype Diagram (AD). Good overviews of these diagramming types can be found in Lane (2000) for CLD and SFD, Morecroft (1982) for CLD, SFD, and SSD, and Sterman (2000) for all except SSD. In this section, we will discuss the CLD, SFD, SSD, and Bull's Eye Diagram, and the way in which an SFD can be translated into a CLD.

6.4.1 Causal Loop Diagrams

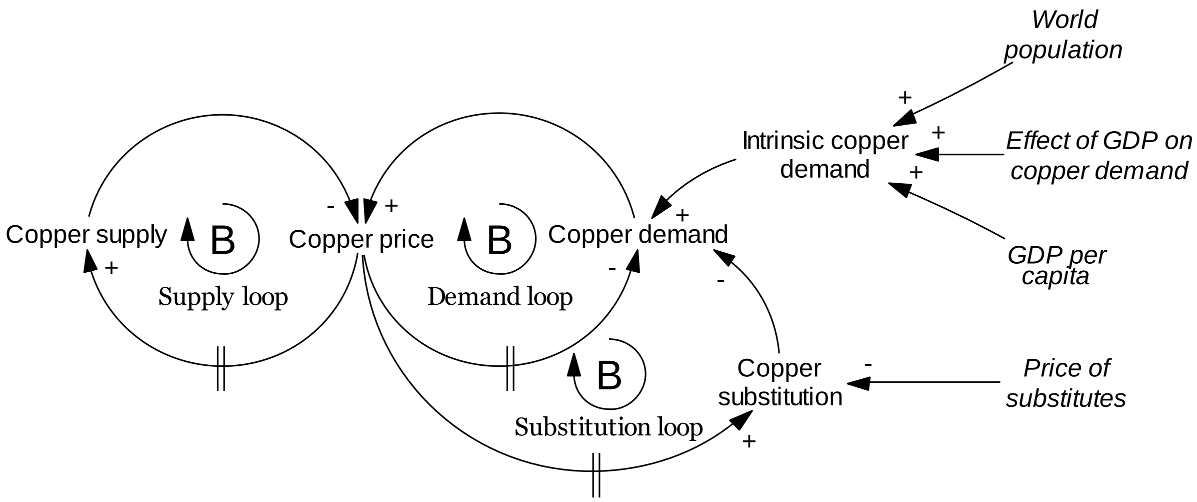


Figure 6.1. Example of a Causal Loop Diagram (from Auping, 2018)

System dynamicists mostly communicate system structures with feedback loop systems by means of CLDs like the one in Figure 6.1. The core building blocks of CLDs are variables and the direct causal relationships between them. These relationships are either positive or negative. However, the meaning of the terms positive and negative does not correspond to their use in everyday life (Sterman, 2000).

A link between two variables A and B is considered positive if:

1. an increase in A causes B to rise above what it would have been otherwise, and
2. a decrease in A causes B to fall below what it would have been otherwise.

A link between two variables A and B is considered negative if:

1. an increase in A causes B to fall below the value would have had otherwise, and
2. a decrease in A causes B to rise above what it would have been otherwise.

The polarity of causal relations is visualized in CLDs by means of link polarities (+ or –) or by means of qualifications of the direction of the cause-effect relation (*S*ame or *O*pposite):

- $\rightarrow^{+}$ and $\rightarrow^{s}$ stand for a positive causal influence: an effect in the same direction;
- $\rightarrow^{-}$ and $\rightarrow^{o}$ stand for a negative causal influence: an effect in the opposite direction;

- delayed relations are depicted by a double stripe (||) through the arrow (e.g., see the arrow in Figure 6.1 between *Copper price* and *Copper substitution*).

The most important goal of CLDs is to identify feedback loops and communicate about them. There are two types of loops or loop polarities: reinforcing and balancing. The loop polarity is determined as follows. If the net effect of the causal links in a loop is negative, the entire loop is negative or balancing (– and B). Otherwise, a loop is positive or reinforcing (+ and R). Be consistent in either calling loops balancing and reinforcing (indicated by B and R), or negative and positive (indicated by – and +).

A simple way to determine the polarity of a loop is thus to count the negative signs. If the number of '–' or 'o' signs in the feedback loop is uneven, then the feedback loop is negative, and if the number of '–' signs in the feedback loop is even, then the feedback loop is positive or reinforcing. The net effect can thus be determined by multiplying the signs of all connections in the loop.

In creating the diagram itself, it is necessary that all loops in a CLD can be unambiguously identified. For this purpose, all loops need to have a name and a loop sign (a circular arrow around either the + or R, or – or B, indicating the direction of the loop in the diagram). If + and – are used to identify reinforcing and balancing loops, the loop names are placed below the loop sign. If R and B are used, loops are generally numbered.

A feedback loop does not contain the same variable twice, except for the first variable. And every feedback loop contains at least one stock or delay variable. The stock variable or delay variable functions as a memory in the loop, and, therefore, helps to avoid simulation problems caused by simultaneous equations, that is, mutually dependent equations that need to be calculated at the same time without starting point. Note, however, that traditional CLDs do not distinguish stock variables whereas hybrid CLDs (i.e., CLDs combined with elements from other diagrammatic conventions) do. See the box for useful guidelines and diagramming conventions to be respected when drawing CLDs.

Guidelines for drawing CLDs (cf., Sterman, 2000, pp. 135-190).

- Iterate;
- Use nouns or noun phrases with a clear (positive) sense of direction as variable names. Choose variable names that, together with the causal links and link polarities, enable users to easily 'read the loops';
- Do not use or conjugate verbs in variable names. The arrows with their polarities perform the role of verbs when reading a CLD;
- Links between variables must be causal and direct, not correlational nor indirect;

- Unambiguously label link polarities (split out links into different effects if polarities are ambiguous);
- Links should be drawn and interpreted under the ceteris paribus assumption (i.e., that everything else remains the same);
- Links are relative: they say that the value of the variable will be above or below what it would have been without the effect;
- Explicitly include the goals of goal-seeking loops;
- Distinguish between actual versus perceived conditions;
- Indicate and unambiguously label loops by loop signs with their polarity and a loop name;
- Indicate important delays on causal links with delay marks;
- Use curved lines for causal links, try to make important loops circular, and minimize crossing causal links;
- Make different types of CLDs for different purposes, audiences or uses (e.g., conceptualisation, loop analysis and communication), and at different levels of aggregation;
- Choose the right level of aggregation (but never too detailed), dependent on the intended use, goal and audience;
- Use SD software to redraw your diagrams;
- For communication purposes, do not use too large a diagram with too many loops;
- And again: iterate.

6.4.2 Stock Flow Diagram

SD simulation models are constructed using Stock-Flow Diagrams (SFDs) as in Figure 6.2. SFDs may also be qualitative and used for communication purposes. In those cases, they are generally simplified or “cleaned” versions of models, which exclude explicit initial values, auxiliary variables, and causal links. Examples of such SFDs are frequently found in literature on modelling of infectious diseases (e.g., Figure 6.3).

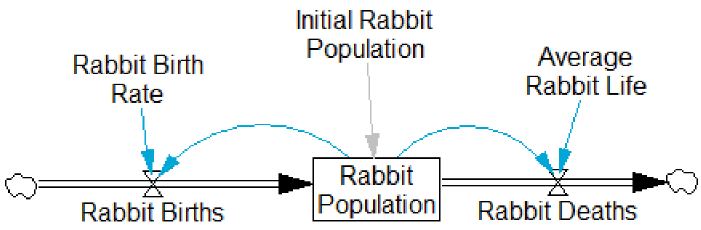


Figure 6.2. Example of a Stock-Flow Diagram in a model

SFDs consist of stock or level variables (☐ , with the stock name inside or underneath the box) and flow variables (⇒). SFDs may contain auxiliary variables (no symbol or O), parameters and constants (no symbol, or in some packages ◊ or O), causal links between variables (→±), and causal links with delay signs (↛±). Finally, SFDs often do not use other symbols (e.g., loops signs, link and loop polarities, and loop names).

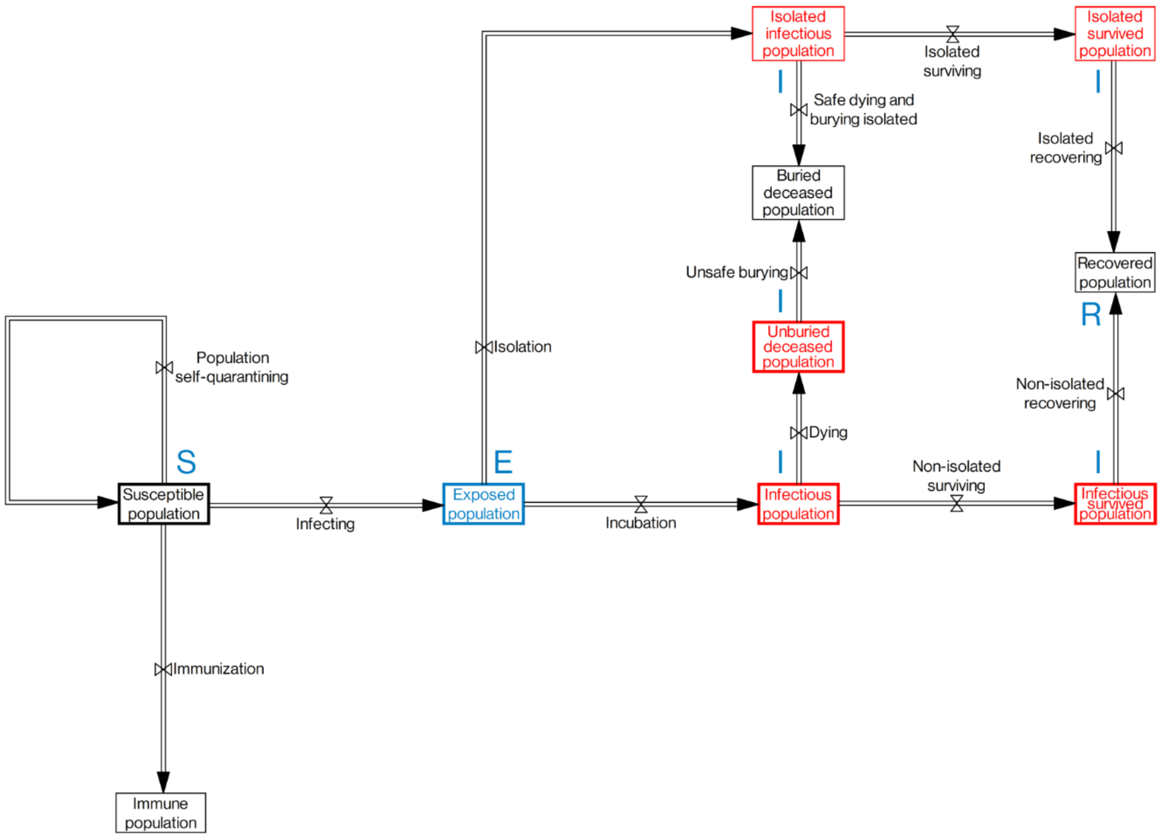


Figure 6.3. Extensive SFD for communication purposes. All causal links, initial causes, and auxiliary variables have been deleted in this figure, but were present in the simulation model (from Auping, Pruyt, & Kwakkel, 2017). The letters S, E, I, and R indicate traditional elements of disease transmission models and do not have any formal function in the diagram.

As Meadows (2008, p. 23) puts it:

> *A **stock variable** – also called a level or a state variable – accumulates, i.e. integrates flows over time. Metaphorically, a stock variable could be seen as a 'bathtub' or 'reservoir'. During simulation, a stock variable can only be changed by ingoing and outgoing flow variables (also called rates). A stock can be increased by increasing its inflow rate as well as by decreasing its outflow rate. Stocks generally change slowly, even when the flows into or out of them change suddenly. Therefore, stocks act as delays or buffers or shock absorbers in systems.*

There are two types of **flow variables**: inflows and outflows. Positive inflows increase the content of the reservoir, while negative inflows decrease it. Positive outflows decrease the content of the reservoir, while negative outflows increase it. Thus, ingoing flows correspond, metaphorically speaking, to taps or valves, and outgoing flows to drains. Flow variables regulate the states of stock variables. Flow variables are thus the variables that need to be targeted by strategies to improve the problematic condition or state of the more inert stocks variables. Stock variables also allow inflows and outflows to be decoupled from each other and to be independent and temporarily out of balance with each other (Meadows, 2008, p. 24).

One way to distinguish between stocks and flows is to imagine what would happen if the entire system suddenly stopped. Flows would then become nil or would cease to exist, while stocks would equal their values prior to halting the system. Technically speaking, each SD model can built with stocks, flows and causal links only. In practice, SD models contain many auxiliary variables, because good SD models are understandable, transparent, glass-box models in which all variables correspond to real-world elements or concepts. Lumping all auxiliaries into the flows would result in opaque models with equations that are too complicated to understand. It is also important to explicitly model all parameters, constants and initial values. Hiding them in auxiliaries and flows makes models opaque and more difficult to use.

Guidelines for drawing SFDs

- Explicitly represent important stock-flow structures;
- All stocks have a rectangular box;
- All flows have a double arrow and a valve sign;
- Stocks are only influenced by flows (or potentially initials, if unambiguously clear);
- Initial values should be made explicit as separate variables;
- Causal relations have single curved arrows;

- Causal relations and may have delay marks, flow may not have these;
- In diagrams, it is generally not necessary to include all causal relations or auxiliary variables which are present in the model. You may even limit yourself to the relevant stock flow structures without including any auxiliaries;
- Do not be afraid to redraw and restructure SFDs: it often takes several iterations to develop decent SFDs;
- Avoid using unnecessary symbols (e.g., polarities) except for delay marks;
- Link successive stocks by means of flows, but only if they are successive accumulations of the same;
- Ageing chains, supply chains, etc. are easier and better represented using SFDs than CLDs; make hybrid diagrams if necessary;
- Do not excessively reduce SFDs: variables should still relate to real-world elements or concepts and should be understandable and useful;

  If you need or want to discuss a model in detail, for example, in a modelling report written for an audience of modellers, then first show the overall SFD with clear sub-models for a big picture overview, followed by well-organized SFDs of each sub-model to be discussed;
- Use the Hide and Unhide tools to hide or unhide parts of the model that are not necessary for the purpose at hand, for example, for explaining the link between structure and behaviour for a specific type of behaviour.

6.4.3 Sub-System Diagram

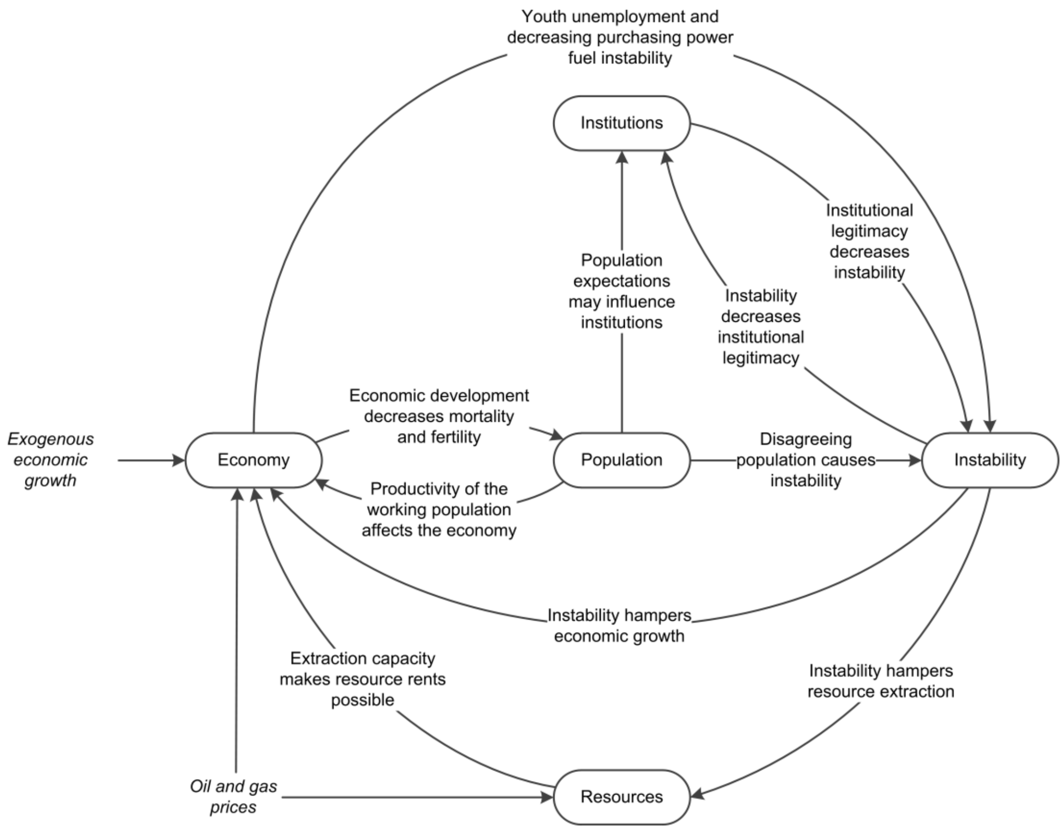


Figure 6.4. Example of a sub-system diagram

Sub-System Diagrams (SSDs) are aggregated diagrams that focus on the main sub-models and the main relations between them, not on the relations within each of them (Morecroft, 1982). SSDs are ideally suited to provide a big-picture overview of issues or systems, with subsystems of variables with many relations between the variables within each of the sectors and few relations with variables in other sectors. Figure 6.4 shows an SSD of systemic factors that contribute to societal instability. The corresponding SFD of the model is too detailed for a big-picture overview. And although CLDs can be used to illustrate various effects, for an overview of the whole model they would be too complex.

Guidelines for drawing SSDs

- Place the names of the sub-systems in rectangles with rounded corners;
- The relation between different sub-models is shown by curved single arrows with text explaining the relation;
- There are no polarities and loop signs;
- Exogenous influences can be linked by straight arrows without text, linking the exogenous influence to one or more sub-systems.

6.4.4 Bull's Eye Diagram

Bull's Eye Diagrams provide a general overview of the elements modelled endogenously, partly endogenously or exogenously, and of all elements that were deliberately excluded from the model scope. Moreover, BEDs allow others to identify unintentional omissions at a glance. There are different ways to structure a BED (see Ford, 1999; Pruyt, 2013). The BED in Figure 6.5 shows for example in a very simple way what is, to various degrees, included in and deliberately omitted from a large and complex simulation model of energy systems (based on research of Kubli & Ulli-Beer, 2016).

It should be noted that BEDs are quite useful for modellers in the conceptualization phase, but less effective to communicate a final model structure to clients and stakeholders. After all, the diagram contains information about which elements are incorporated in the model, and to what extent they are exogenous or not, but does not include the relation between the variables. Therefore, a BED is not always suitable for communicating the system structure, which is, after all, caused by the relations between the variables in the model. The use of BEDs is thus mostly limited to scoping (which may be part of a joint sense-making process between modellers and stakeholders) and model documentation.

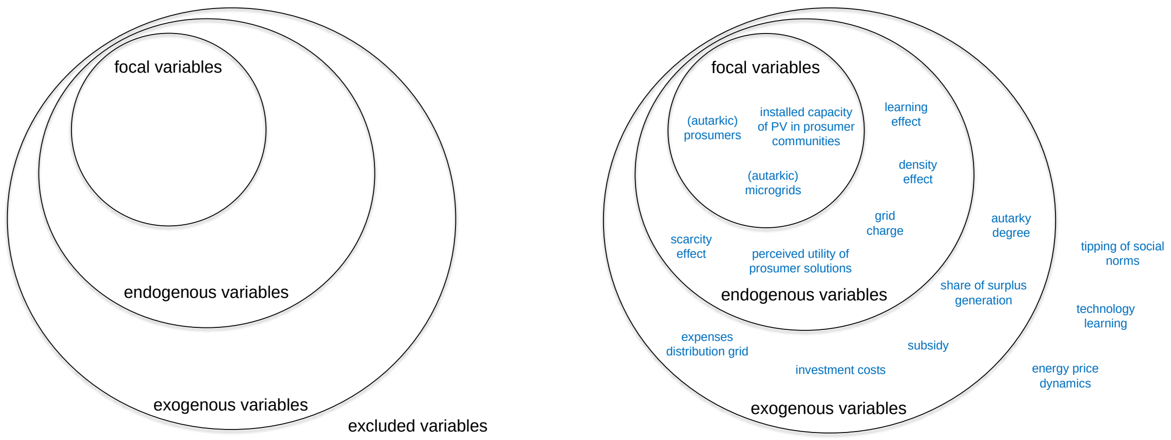


Figure 6.5. General example of a BED (left) and a BED specific to the effect of consumer energy production on the energy system (right).

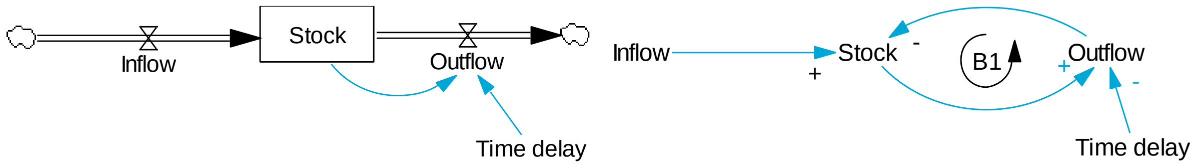
6.4.5 Translating SFDs into CLDs

SFDs can be directly translated into CLDs. The first important relation in that respect can be found in the definition of a stock: *Stock = Inflow – Outflow*. A stock is thus positively influenced by an inflow, and negatively by an outflow (Figure 6.6).

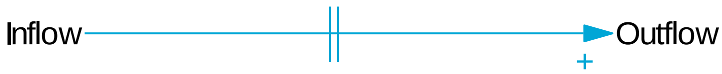
(a) (b)

Figure 6.6. Translation of an SFD (a) into a disaggregated CLD (b).

In this manner, a *disaggregated* CLD can be made of every SFD. Do note that in this case a decrease in the *Inflow* will not cause a decrease in the *Stock*, but only a smaller increase.



(a) (b)



(c)

Figure 6.7. Relation between an SFD with a delay structure (a), a disaggregated CLD (b), and an aggregated CLD (c).

In communications on SD research, disaggregated CLDs are rarely used. Instead, *aggregated* CLDs are more common. Aggregated CLDs make use of the fact that, by definition, *Outflow* which is defined as *Stock / Time delay* is a first-order delay of the *Inflow* of that same stock (Figure 6.7a). The disaggregated CLD of that same structure (Figure 6.7b) has a balancing loop between the *Stock* and the *Outflow*. This structure, or similar, can be translated into a single causal link between *Inflow* and *Outflow*, sometimes made explicit with a delay mark (Figure 6.7c). Figure 6.7c is thus an aggregated CLD of the SFD shown in Figure 6.7a.

The basic rule in making aggregated CLDs is that directly connected stocks and flows should not be included together. This means that in connected stock-flow *structures*, like the ageing chain in a population model or the different phases of disease infection in a transmission model, you choose to either show the stocks or the flows. The choice is often based on which variable is most important for explaining the model's structure, given the purpose of the model. The choice may also be informed by trying out different variants and seeing which generates the most elegant picture of the structure.

When making *highly aggregated* CLDs, the goal is to reduce the number of variables further to the absolute essence of the structure that explains the behaviour of the model. You would do this when making a CLD about a project model. This is more an art than a science. While the translation of a SFD into a disaggregated CLD is an exact science (there is exactly one solution), making aggregated CLDs is based on tacit knowledge of what to leave out and what to include. For highly aggregated CLDs, this applies even more. This knowledge can only be mastered by practicing a lot.

6.5 The Dynamic Hypothesis

The dynamic hypothesis is a working theory of how the problem arises from the structure of the system (Lane, 1993; Randers, 1980). It forms the link between the diagram used to conceptualise the system and a reference mode of behaviour. In formulating the dynamic hypothesis, you therefore do well to refer to structure elements, like feedbacks, delays, and accumulations, that cause specific system behaviour.

The goal of the model analysis that occurs in a later phase can be seen as an attempt to falsify the dynamic hypothesis. For example, in research on the geopolitical impact of the US shale revolution, a new system of drilling that increased the use of oil and gas in US (Auping, Pruyt, De Jong, & Kwakkel, 2016; De Jong, Auping, & Govers, 2014), the researchers hypothesised that the shale revolution would cause gas for gas substitution (i.e., natural gas replaced by shale gas) both in the US and Europe. This was assumed to lead to significantly lower gas prices, but to affect oil prices only in a limited way. Running the model showed, however, that the accumulation of oil (a stock) would have a larger effect on the global energy market than the availability of natural gas, which can only be stocked to a limited amount and is thus sold from the pipeline (a flow). The model runs showed that the system structure made it more plausible that oil prices would come under pressure instead of the natural gas prices, and that this would have a large effect on government finances in rentier states around Europe. This research was performed and published before the oil prices collapsed in 2014, a period when most analysts believed that the oil prices could only rise.

6.6 Qualitative analysis and system archetypes

By analysing a conceptual model, information on the system can be obtained. This section is intended to give some understanding of the qualitative analysis of causal SD models and to provide knowledge about frequently occurring behaviour in various types of systems. We will first of all look at some simple structures, followed by some well-known and more complex structures, the so-called archetypes.

6.6.1 Stability and causal diagrams

The behaviour of a (simple) autonomous system, in which no time-dependent influences from the environment affect the system, can be stable or seek equilibrium, show oscillations, or even show continuous autonomous growth. For simple systems it is also possible to establish indicative relations between the system structure and system behaviour without accurate mathematical formulations. A few examples can illustrate this.

Example: capital and interest

A starting capital is put into a bank account at a fixed interest rate. We are interested in the development of the capital as a function of time. The relevant variables are the *size of the capital* and the amount of *interest* added per time unit. The amount of *interest* is in proportion to the *capital*. As the *capital* becomes larger, the *interest* received will be larger; and as the *interest* is larger, the *capital* grows faster. Thus, both influences are positive and because they form a closed loop. This is called a positive feedback loop.

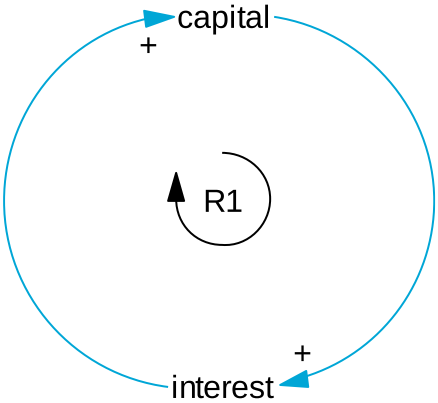


Figure 6.8. Causal diagram of a positive feedback loop

Such growth behaviour is typical for all dynamic systems with one positive, linear feedback as the basic structure. This is called a reinforcing loop. Examples include an economic system with a constant growth rate per year, or a population system with a constant growth percentage.

Simple systems with a single, negative feedback loop without a time delay usually show a tendency towards stable behaviour. However, generalisations about the behaviour of these loops must be viewed with some caution, as we shall see below. Negative loops are called balancing loops.

Example: water tank

A water tank has a constant *inflow*, while the *outflow* depends on the level in the tank. As the *level* goes up, the *outflow* becomes larger, which in turn has a negative effect on the *level*. With a constant *inflow*, this system tends towards a state of equilibrium and is, therefore, stable. This causal structure is sketched in Figure 6.9.

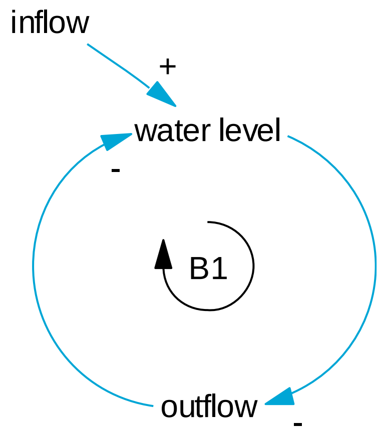


Figure 6.9. Causal diagram of a negative feedback loop.

Negative feedback is also shown to lead to stabilising behaviour in a number of other systems, such as a mass-spring system or a pendulum. In some economic processes, negative feedback also occurs. The classical example is the so-called hog (hogs are domesticated pigs) cycle (Hanau, 1928) discussed below (Figure 6.10).

Example: hog cycle

When pork prices are high, pig breeders will buy more piglets. If these *pigs* are put on the market once they have grown, the *supply* goes up; if the demand stays the same, the *price* will drop. As a response, the pig breeders will breed fewer *piglets*, and the (negative) feedback loop is closed. However, if the responses are too strong, the system will show strong fluctuations.

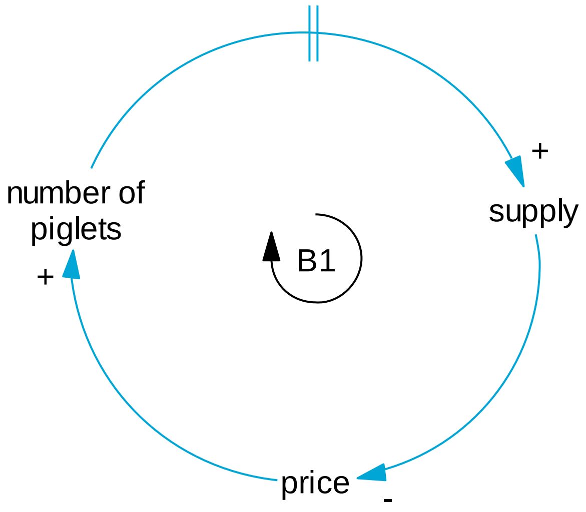


Figure 6.10. Causal diagram of the pork production cycle

Negative feedback occurs in many processes controlled or invented by man (but also occur in nature). Examples include a supermarket where the manager will open more checkouts if there are long queues in order to reduce the waiting time: a negative feedback intended to limit the waiting time as much as possible. Another example is stock management, where departures from the desired number of stock may lead to additional buying (if the stock is too low) or less buying (if there is too much stock).

However, caution is needed, since negative feedback in a system does not always lead to stable behaviour. Negative feedback may also lead to instability in complicated systems of a higher order, or in systems with a delay.

In reality there are few systems that are characterized by only one type of feedback loop. Usually there are combinations of positive and negative feedback.

Example: population

In an autonomous development of a population, the basic structure is as sketched in Figure 6.11.

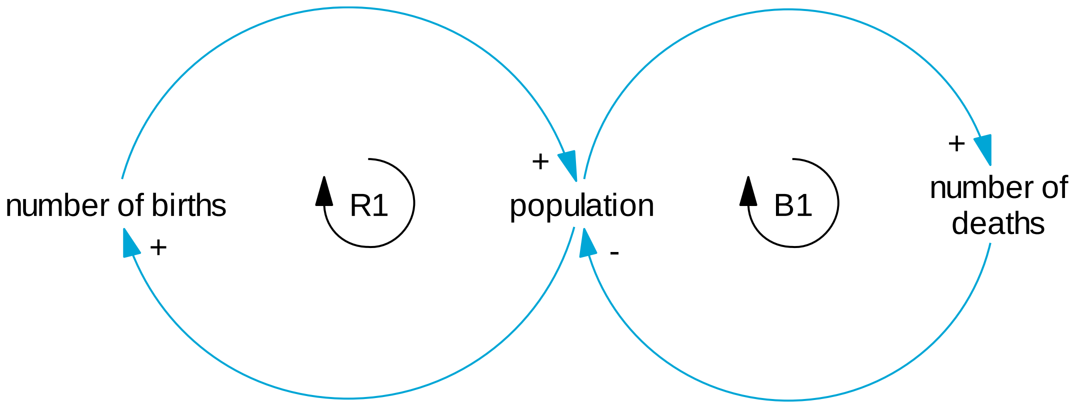


Figure 6.11. Development of population

The development of the *population* as a function of *time* is determined by two feedback loops: a positive feedback loop (the larger the population → the larger the number of births per unit of time → the larger the population) and a negative feedback loop (the larger the population → the larger the number of deaths per unit of time → the smaller the population).

The behaviour of this system is determined by the ratio between the two loops: if the positive loop dominates (the birth rate is consistently larger than the death rate), growth occurs; if the negative loop dominates (the death rate is consistently larger than the birth rate), a decrease towards an equilibrium occurs – in this case 0 – and the system consequently can be called stable.

In practice, shifts in the ratio between the birth and death rates will occur over time, for example because a shortage of food occurs or wars break out when the population gets too large. In that case, growth can change to decrease and the system tends towards an equilibrium, for example depending on the food stock.

The behaviour of most systems is the result of complex interactions of positive and negative feedback, which sometimes exist naturally and sometimes have been deliberately added by man. For now, we will suffice with the general observation that

– in simple cases – the nature and strength of the relations can be estimated from the system's behaviour and the nature of the relations between the system variables, in particular the presence of positive and/or negative feedback loops. However, more accurate statements about the system's behaviour require us to determine the relations more accurately.

6.6.2 System archetypes

Some particular, generic categories of qualitative system structures are called system archetypes. These are simple, small combinations of one to a couple of feedback structures that act together in causing particular system behaviour. In his book on systems thinking, *The Fifth Discipline*, Senge (1990) discusses archetypes that will be described in this section on the basis of the corresponding causal diagrams. They are the analyst's tool to learn across problem domains in two ways.

First, the system archetypes serve as prospective and diagnostic tools. In this way, system archetypes offer insight in underlying structures for problematic system behaviour. Looking at real-world problems, trained systems thinkers can recognise common patterns of behaviour, across systems and domains. Additionally, the system archetypes can help us to understand how situations can be improved, for instance by identifying policy levers that typically work in similar situations.

Second, these system archetypes can also be used as a generic part of a conceptual model. System archetypes offer a scope and a basic structure within which a model can be further developed or constructed, which may be helpful when you undertake your own simulation modelling study.

The archetypes we discuss here are the following: balancing process with delay, limits to growth, shifting the burden, eroding goals, escalation, success to the successful, tragedy of the commons, fixes that fail, and growth and underinvestment. Each archetype has so-called 'early warning' symptoms and there are several ways to deal with the resulting situations. These are described for each archetype.

Balancing process with delay

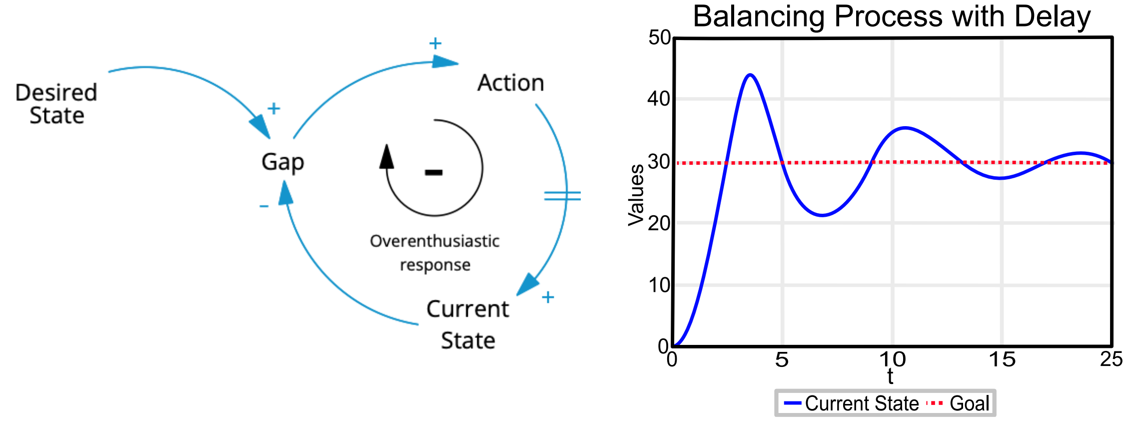


Figure 6.12. Archetype diagram and possible behaviour of a system with delay.

The ‘balancing process with delay’ archetype relates to systems characterized by regulation based on delayed feedback. Corrective interventions are often too strong and fast for the system, thus causing oscillatory behaviour or insufficient results.

A person, group or organisation working to achieve a particular objective will adjust the behaviour on the basis of delayed feedback. If they are not aware of the delay in the system, more corrective action than necessary is taken, or they give up because they do not see any progress. In a slow system, action that is too aggressive can lead to instability, for example oscillations with an increasing amplitude. The beer game (see www.beergame.org) is one example of this type of system; other examples include a shower in which the water temperature only slowly responds to changes in the tap position. Yet another example can be found in housing: project developers keep building new houses until the demand goes down, but by that time so many houses are already being built that the market can collapse. Conversely, consider the balancing process of a system with no delay, or only a very small delay. What would happen? The system behaviour would approach stability. In this case, and many other cases, the delays determine the system behaviour.

Situations that are characterised by a balancing process with a delay typically trigger the following actor reaction: ‘We thought we were in balance, but overshot the goal.’ To prevent this, one could adjust the delay times, for example by deciding to wait longer before making correcting interventions, using less strong interventions, or making the system more responsive.

Limits to growth

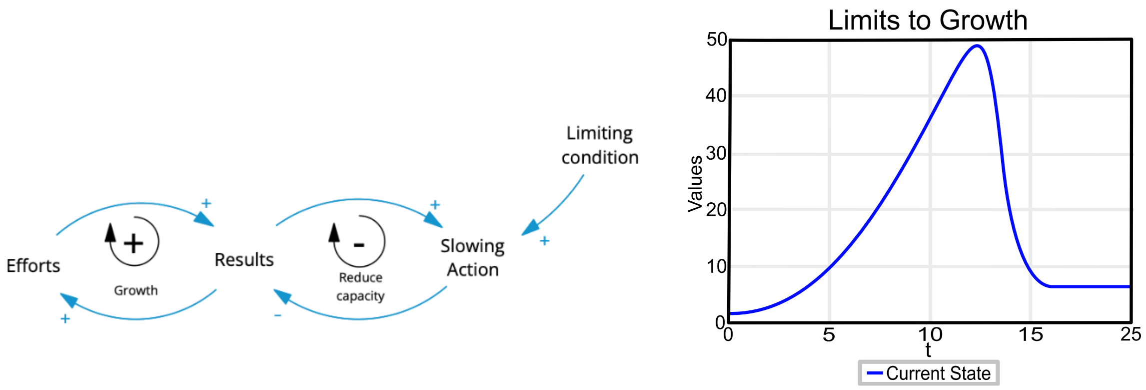


Figure 6.13. Archetype diagram and possible behaviour of a system with limits to growth.

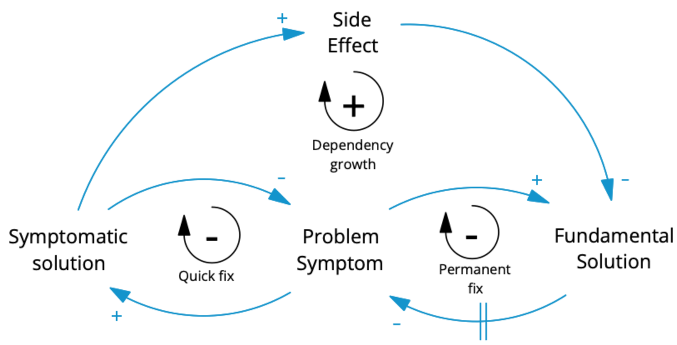
The ‘limits to growth’ archetype (sometimes called ‘limits to success’) describes situations in which growth is followed – after reaching a limit – by stagnation, and possibly by collapse. Any system with a carrying capacity or limiting condition can experience limits to growth.

A system behaves in such a way that growth occurs during a certain period, after which this growth starts to stagnate and then stops. This may lead to a collapse. The growth phase is caused by one or several positive feedback loops. The slowing is caused by a balancing process, which starts if a particular limit is reached. Collapse takes place if the negative feedback loop becomes more dominant than the positive feedback loop.

A well-known example is an animal population that starts to grow quickly after its natural enemy has been removed, reaching and then crossing the system's carrying capacity, until the animals overgraze their land and collectively die from starvation. Another example is a town that continues to grow until the existing area has been filled up, which leads to higher land prices, limiting further population growth.

The following question is often asked in reaction to early symptoms: 'Why worry about problems that are not there?' To deal with a situation in which limits to growth exist, one can accept the limits, remove the limiting conditions and/or balancing feedback loops, and/or identify potential balancing processes before they begin to affect growth. Removing or decreasing the restrictive conditions, for example, by importing food or reclaiming new land area, may ensure that the growth process continues, if this is desired.

Shifting the burden



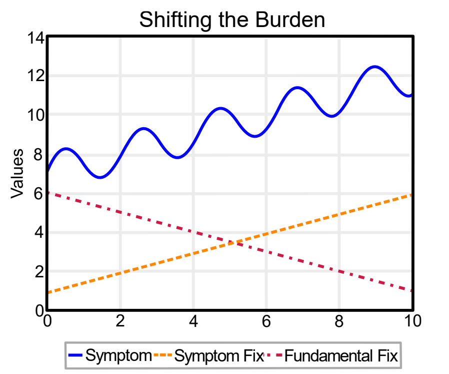


Figure 6.14. Archetype diagram of shifting the burden.

The 'shifting the burden' archetype relates to short-term 'solutions' that provide immediate and positive results. However, as this solution is used more frequently, more fundamental long-term corrective measures are used less frequently and after some time, the capabilities to achieve fundamental solutions may be rendered entirely inoperative. This archetype is sometimes called 'addiction', 'dependence', or 'shifting the burden to the intervener'.

An example is when a company hires a consultant to solve a problem by treating the symptoms, but not teaching the employees to handle problems themselves in the future, thus losing the opportunity to acquire skills in-house. A similar critique is often voiced about international development aid. For example, when Western philanthropic organizations imported free drinking water wells (produced in Asia) to West-Africa in the nineties, they eliminated the use and upkeep for existing wells. When these cheap, imported wells started to break down, there was no maintenance infrastructure and all local knowledge on well maintenance was gone, exacerbating the initial problem. Another example of ‘shifting the burden’ is trying to sell more goods to existing customers instead of expanding the number of customers. Paying bills by borrowing money instead of economizing may also lead to this behaviour.

In all these cases, the initial solution seems to be effective (e.g., problems are solved, drinking water is available, more income is generated, bills are paid). So why should anything change? ‘Shifting the burden’ and addiction problems are very challenging to resolve, so preferably you should avoid these problems entirely. To prevent problems of this archetype, one should stay focused on long-term re-structuring, analyse the consequences and side-effects of potential actions thoroughly, and beware of virtue signalling or symbolic measures.

Eroding goals

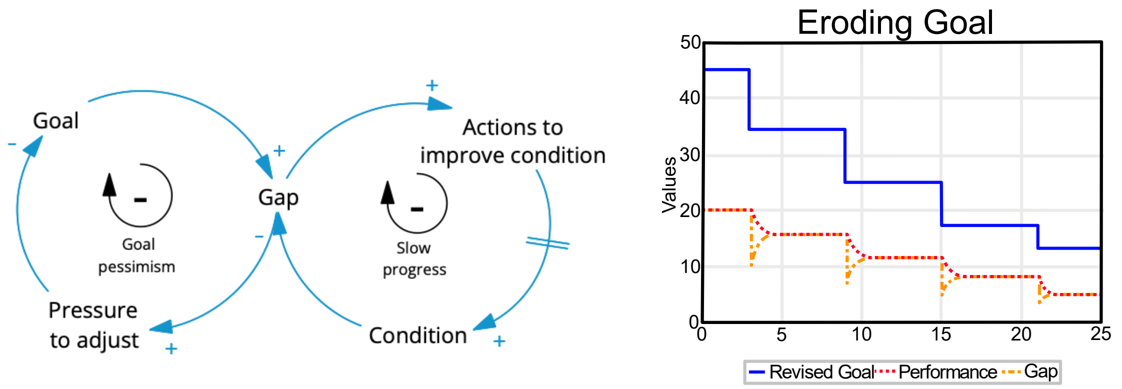


Figure 6.15. Archetype diagram and typical behaviour (with discrete adjustments of the goal) of eroding goals.

The ‘eroding goals’ archetype describes situations in which short-term solutions lead to adjustment of long-term goals. These systems just keep getting worse. Other terms for this archetype are ‘the boiling frog syndrome’ or the ‘drift to low performance’. This archetype looks similar to the previous archetype (shifting the burden), but in the ‘eroding goals’ archetype, the goal is allowed to slip. In other words, the gap between the desired situation and the current situation is reduced by desiring less.

An example of ‘eroding goals’ is a company that decreases the quality standard through economizing, instead of investing in qualitatively better production methods, while assuring that quality is maintained. Other examples can be found in continuously lower air quality standards or adjustment of sustainability goals. In such a situation one usually thinks that it is suitable to lower the standard for a short period, because things will turn out better after some time. It is, however, better to stick to long-term vision and goals.

To repair problems that fit into the ‘eroding goals’ category, one should either attempt to keep standards absolute, or make benchmark goals to the *best* performances of the past, instead of the *worst*.

Escalation

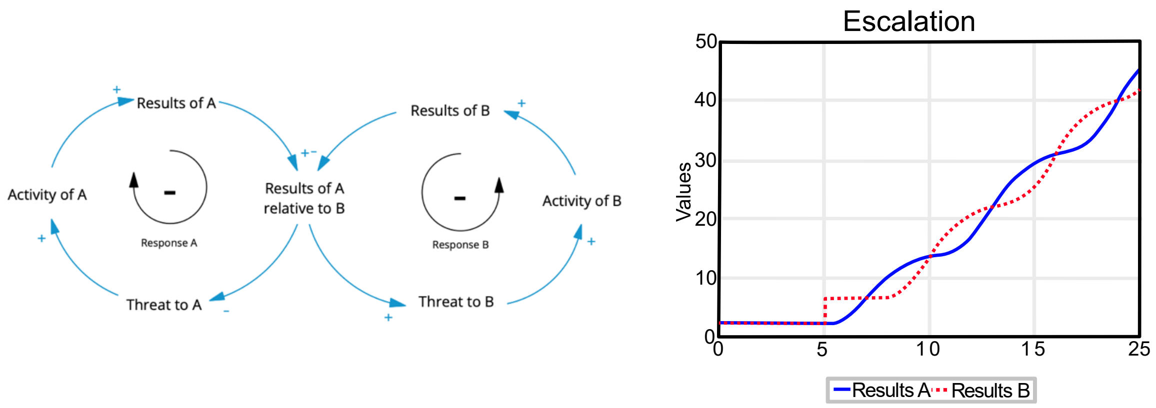


Figure 6.16. Archetype diagram and possible behaviour (after an initial perturbance of B’s results at t=10) of escalation.

The ‘escalation’ archetype describes situations in which goals are dependent from the (perceived or real) status of other actors. In these situations, two or more parties aim for relative advantages over the other party or parties, resulting in an escalation. Note that the two negative loops in Figure 6.16 jointly form one positive loop, explaining the escalating behaviour.

In this system archetype, two actors regard their welfare as depending on a relative advantage over the other. If one of the two gets an edge, the other feels threatened, which leads to more action to improve that person’s own relative position, after which the first one feels threatened, etc. This results in an escalation and may cause unstable behaviour. This type of situation can for example be seen in advertising campaigns or in an arms race. In these situations, we may say: ‘An eye (and something more) for an eye’. Note that positive forms of escalation can be good, for example the race to a cure an illness or the race to be the first on the moon.

Whether positive or negative, it is notoriously difficult to stop such escalations, so one should avoid getting into this trap. Indeed, in this type of situation the opponent is often blamed for the outcome, saying: ‘If they stop, we will stop too’. Generally, the only ways out of an arms race are 1) unilateral disarmament (one person has to stop first) or 2) disarmament agreements, for instance by looking for win-win situations.

Success to the successful

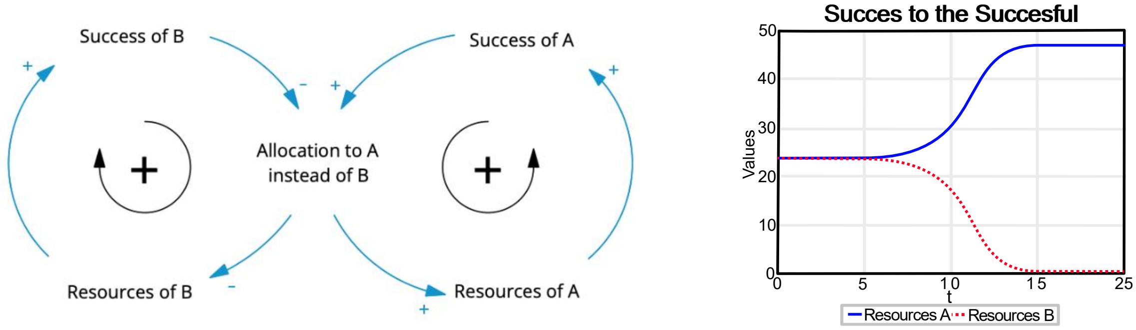


Figure 6.17. Causal diagram and possible behaviour of the situation in which the successful are becoming more successful.

The ‘success to the successful’ (or: ‘competitive exclusion’) archetype describes situations in which resources are accumulated by single actors. Situations characterised by this archetype involve a minimum of two interacting, equal-status actors, of which one will stay behind even though they have the same attributes. These parties or activities compete for the same limited resources and even a small advantage results in more resources being allocated to the most successful party or activity, which reinforces the competition.

Wealth, information and privilege are examples of such coveted resources. As such, the ‘success to the successful’ archetype is at the core of economic discourse, see for example Marx’ critique on capitalism, or more recent discussions about the wealth gap between billionaires and others. Another example of such a multi-actor situation is a company in which two products consume a certain amount of financial investment and attention from management. One of the products is a success on the market, so that more is invested in this product, which means that less money is available for the other product. Moreover, the board game ‘Monopoly’ shows how players who do well in the earlier rounds have a higher chance to bankrupt their opponents.

The resolutions for the ‘success to the successful’ archetype can be individual and societal. Individually, the stay-behind can diversify: they try to fix their position by using different resources or other measures of success. Societally, bodies of authority can install policies that level the playing field, such as policies that create rewards for success that do not bias the next round of competition, and policies that put limits on monopolies or dominance (e.g., antitrust laws). There exist many devices that aim to break the loop of the richer getting rich and the poorer getting poor, including charity, gift-giving and public welfare.

Tragedy of the commons

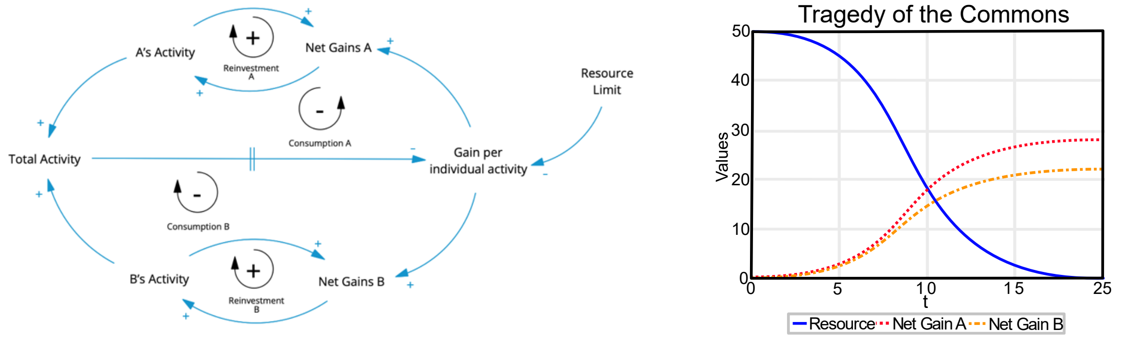


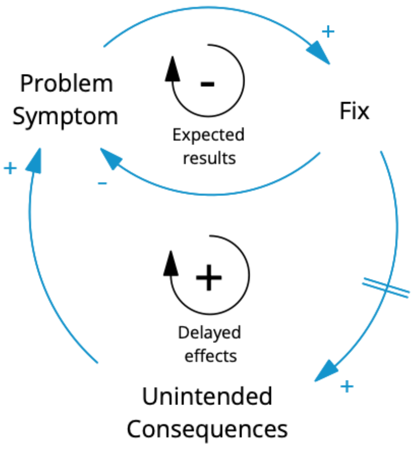
Figure 6.18. Causal diagram and possible behaviour of the situation in which a common resource runs out.

The 'tragedy of the commons' archetype describes situations in which individual incentives lead to catastrophic dynamics for the collective. It describes systems with growth or escalation in a commonly shared, erodible environment, and where the number of users increases at a rate not influenced by the condition of the commons (as opposed to 'limits to growth'). Moreover, missing or heavily delayed feedback from the resource to its growth hinders adaptation in these situations.

In this case, actors use a common resource that is available to everyone but is limited in amount. First, they are rewarded for using the resource. Additional use results in a separate profit for each of them, but at a certain point, less and less of the resource is available, resulting in less profit, followed by intensification of the activity. Ultimately, the resource can become depleted. Examples of these situations are depletion of resource systems such as over-fishing, over-grazing or running out of raw materials.

The 'tragedy of the commons' archetype makes people think that there is more than enough of the resource for everybody; a long-term vision is lacking. There are three general ways to govern the commons: 1) regulate the commons (e.g., by quoting), 2) appeal to the actors' morality by education and exhortation (e.g., self-regulation, peer pressure), and 3) privatization of the commons, under the assumption that people have more self-control to stay below the carrying capacity of their own private resource, as they harm themselves in a more direct way (Meadows, 2008).

Fixes that fail



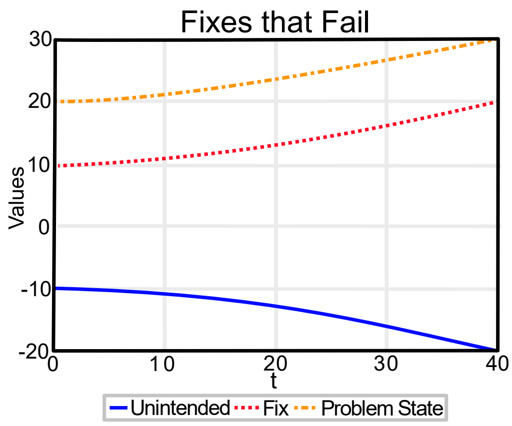


Figure 6.19. Causal diagram of the situation in which a fix results in unintended effects.

The ‘fixes that fail’ archetype describes fixes that result in substantial unintended and undesirable side effects, and as such, instead of solving the issue, create new issues. These fixes often are the result of linear causal thinking instead of system thinking, or of treating symptoms instead of root causes systemically. The archetype is sometimes called ‘fixes that backfire’ and – if implemented policies are never truly effective – ‘policy resistance’. In this situation people wonder why a measure is not working now, as it has always worked before.

A difference between this archetype and the ‘shifting the burden’ archetype (Figure 4.12) is that the unforeseen consequences in the ‘fixes that fail’ archetype result in the problem becoming worse, whereas in the ‘shifting the burden’ archetype the side effects result in a fundamental solution not being used.

As a quick fix is effective in the short term, and only has unforeseen consequences in the long term, the same fix may possibly be used again. An example of such a situation is saving on maintenance costs, which leads to more breakdowns, higher costs and in turn, because of a lack of funds, to more savings on maintenance.

A way to avoid ‘fixes that fail’ is to stay focused on the long term by considering long-term effects and unintended consequences of policies from the outset. If you want to fix an already implemented fix that is failing, one can either overpower, or the counterintuitive opposite: let go entirely, give up ineffective policies and (re-)align actors’ goals. In all situations, one needs to stay focused on the long-term effects and solutions needed to manage this situation.

Growth and underinvestment

This archetype is an extended version of ‘limits to growth’. The difference with ‘limits to growth’ is in this case that the limiting condition can be fixed, but is not.

An example is a company in a growth situation, which makes insufficient investments. Goals may be adjusted to justify the underinvestment. If this happens, the lower goals will lead to lower expectations. Another example of such a situation is a company that hires employees to meet the growing demand, but fails to train the new employees sufficiently. Instead of investing more money in training, it adopts a wait-and-see attitude until the number of customers goes down as a result of dissatisfaction with the company's service. A possible way of handling this might be creating sufficient personnel capacity in advance of the demand. This may also work as a strategy for creating demand.

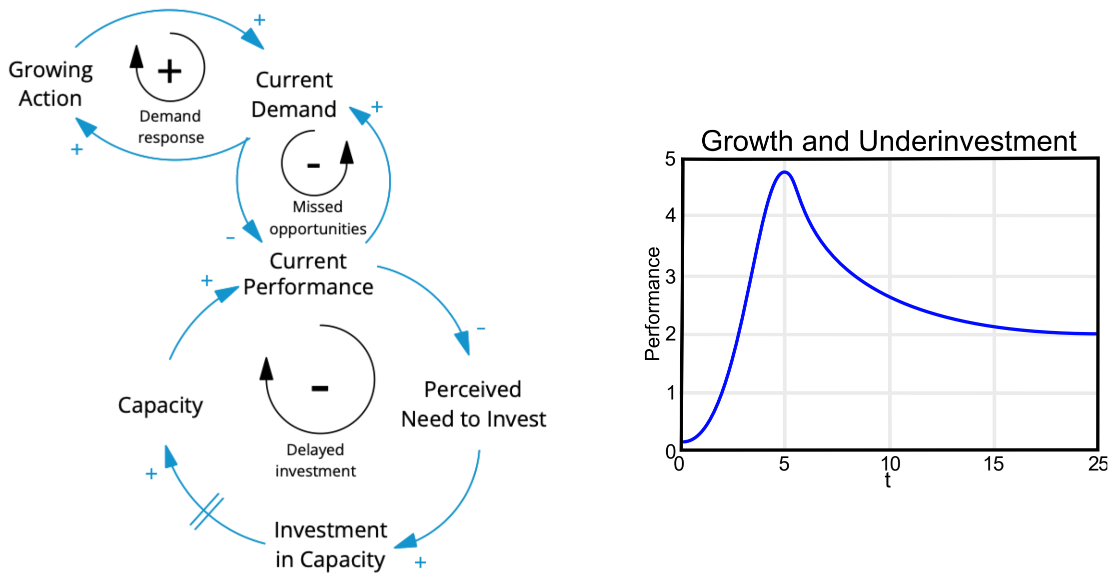


Figure 6.20. Causal diagram and possible behaviour of growth and underinvestment.

A warning sign is if problem owners say: 'We used to be good at this, but let's save money now and not invest too much.' Instead, in these situations it would be wiser to build capacity based on a solid long-term strategy, and hold on to the company's vision.

7. Formulation

This chapter discusses the formulation of System Dynamics models. The formulation phase results in a quantitative model. Formulation means that a conceptual model is transformed into a formal quantitative representation consisting of equations and specified parameters. The quantitative model is the starting point for a more detailed analysis of the system behaviour.

7.1 Numerical methods

In order to solve an SD model on the computer, the user has to select a numerical integration method. The user should therefore be aware of the pros and cons of the various methods and compare them for their trade-offs. The best choice chiefly depends on the formalisations used in the model. Similar considerations apply to the choice of the size of the time step (the delta time).

Several general guidelines that may be useful in the selection of a method and corresponding time step are offered below.

7.1.1 Choice of method

The Euler method

Strictly speaking, the Euler method of integration (Euler, 1768) should be used whenever the *derivative* of any variable in the model is discontinuous. This is frequently the case in SD models. For example, if a MIN or MAX function is used and switches from one argument to the other, then this frequently results in a discontinuity in the derivative. The same counts for LOOKUP functions, IF THEN ELSE logic, or the use of *integer* values.

Only in the case of discrete functions, the Euler method will show better results. For example, if one uses PULSE functions to fill or empty a stock, the Runge-Kutta methods will not generate correct results.

The Runge-Kutta methods

The Runge-Kutta or RK methods (RK, Kutta, 1901; Runge, 1895) – specifically RK2 and RK4, with fixed time step or auto – have been designed for *continuous models*; they cannot handle integer values and large discontinuities in the derivatives of variables. If you use a RK method, remember to check whether there are any discrete variables and whether integer values are generated in the model. Use one of the RK methods, preferably RK4 auto, if the model is continuous and has oscillating tendencies.

Despite this general advice, it must be pointed out that the combination of the time step and the method chosen is essential. For the same step size, Runge-Kutta is more accurate than the Euler method, but Euler with a smaller time step can be more accurate than Runge-Kutta with a larger time step. It is very important that the correct combination of method and time step is selected. If the combination is not accurate enough, the simulations may be erroneous. For instance, if you pick the wrong model, the model can exhibit oscillations of increasing amplitude although in reality, oscillations in the system are damped.

7.1.2 Choice of time step

- Choose a time step that is a good compromise between the accuracy of the result and the speed of calculation.

The smaller the time step, the more accurate the result. Unfortunately, the price of accuracy is that it may take longer to run the model. Choosing the right step size may require a few test steps. As a rule of thumb, a time-step ranging from 1/2 to 1/10 of the smallest time-constant in the system (for example smoothing time, decay time, adjustment time) is a good first approach. If you can trace the smallest time constant, this is a good *starting point* in determining the time step.

- Always test the model for time step sensitivity.

Always start with a full run of the model and plot a few variables, especially those showing capricious behaviour. Then halve the step size and repeat the run. Compare the values of these two runs. If they differ noticeably, repeat the process with the step size reduced by half again. Keep repeating this until there are no more visible differences between two consecutive runs.

7.2 Stocks

Setting up stocks is perhaps the most important part of SD model construction. It is important that you know how to set up stocks properly, as these basic principles allow you to construct models on subjects with which you do not yet have any experience. In many areas of physics, such as mechanics, fluid dynamics and electricity, commonly-known laws that are frequently used in these fields can be derived on the basis of balance equations.

Within SD models, stocks have the function of keeping track of accumulations of physical quantities within the model. Examples include population development and the spread of infectious diseases. However, they also function as memory in your model for quantities more difficult to measure.

In setting up a stock, we look at the quantity (level) of a variable within a particular scope and determine why and by how much this quantity changes as a function of time. The level of the stock is a “state variable” of the model.

First of all, the scope must be defined. The boundary between this area and the environment must be clear. The scope does not have to be a spatial area, but in broad terms can be defined as the domain of interest.

In a stock equation, we are looking for an expression for the change in the level of a variable in during a particular interval. The general expression for a stock equation of a continuously modelled process is:

$$\frac{dStock(t)}{dt} = \sum \big(Inflows(t) - Outflows(t)\big) \quad \text{Eq. 7.5}$$

Or, as an integral:

$$Stock(t) = Initial\ value + \int_{t_0}^{t} Inflows(t) - Outflows(t)\, dt \quad \text{Eq. 7.6}$$

The flows are transports across the boundary of the stocks. The steps for setting up a balance equation are:

1. Select stock;
2. Select scope;
3. Set up the stock equation;
4. Set up flow equations.

When numerical values are entered into a SD model, the sum of all flows to or from a stock must equal the right-hand side of the stock equation, representing the change in the stock as a function of time, i.e. the difference in the stock from one time step to the next.

Example: population stock

- Select stock

If we are interested in the size of a population at a particular moment in time, a population stock can be formulated.

- Select scope

The scope must be determined next. In this case, for example, it could be the Netherlands.

- Set up the stock equation

The population flows going in and out must be determined, in this case how much emigration and how much immigration takes place each year. Finally, the numbers of births and deaths each year must be determined. This results in an equation with the following form:

$$Population(t) = \int_{t_0}^{t} Births(t) - Deaths(t) + Immigration(t) - Emigration(t)\, dt \quad \text{Eq. 7.7}$$

- Set up flow equations

The terms on the right-hand side of the equation are expressed in such a way that they can be calculated. For populations, the annual birth and death rates per capita are usually known. The equations will then be:

$$Births(t) = Population * Birthrate \quad \text{Eq. 7.8}$$

$$Deaths(t) = Population * Deathrate \quad \text{Eq. 7.9}$$

For annual emigration and immigration – depending on the situation – a function can be based on figures from the past, for example, or they can be assumed to be constant. The data required may be retrieved from agencies such as national statistical agencies, or obtained through measurements, or by estimation.

When solving the equations, the software will substitute the flow equations above into the stock equation. You do not have to do this yourself. This results in a differential equation that can be solved if the initial value is known. In this way, the value of the stock can be calculated as a function of time and, for example, a graph of the population as a function of time can be created. The software will thus solve the following equation numerically for each time step.

$$Population(t) = \int_{t_0}^{t} Population(t) * Birthrate - Population(t) * Deathrate + Immigration(t) - Emigeration(t)\, dt \quad \text{Eq. 7.10}$$

This type of equation can also be set up, for example, for the staff size in a company, the quantity of algae in a lake, or the users of social media.

7.3 Flows

Flows are the *only* means of changing stocks. In the previous example, the flows are births, deaths, immigration and emigration. There are few rules for the equations of flows, apart from the fact that **their units are always the units of the connected stock divided by the time unit**.

There are a number of frequently occurring types of flow equations, which can be classified into the following general categories:

1. Flow = coefficient * stock (see 7.3.1.)
2. Flow = (variable - stock) / constant (see 7.3.2)
3. Flow = normal flow + effect (see 7.3.4)
4. Flow = normal flow * effect (see 7.3.5)

This is not an exhaustive list; other situations may occur in practice. These different types of equations will be discussed below, using a number of examples. While studying this section, it is a good idea to create the stock-flow diagrams corresponding to the different flow equations to see how they work.

7.3.1 Flow = coefficient * stock

This type of flow equation can appear in several forms. In its simplest form, the coefficient is a constant.

- Flow = constant * stock

In the equation above, the change in the stock is linearly related to the value of the stock. These situations occur in modelling the size of a population, for example, or in determining the size of a bank account on which interest is received. The amount of extra money paid each month is a certain percentage of the amount of money already in the account.

If this flow is the only effect on the stock, the corresponding stock equation is:

$$Stock(t) = \int_{t_0}^{t} c * Stock(t)\, dt \qquad \text{Eq. 7.11}$$

If we were to solve such an equation analytically as a function of time, an exponential function would result for the stock. This means that such a stock shows exponential behaviour. Always make sure that the dimensions or units of flow equations are correct and that the parameters have a meaning.

- Flow = (1 / constant) * stock

An equation representing an outflow of a stock often has this form. The average *outflow of employees* from a company may be represented by dividing the *number of employees at a certain moment in time* by the *average time employees work for a company* (this is the same as multiplying by 1/(*time worked*). By using the time period in the equation, we ensure that a parameter is used that has a physical meaning. Natural processes of decay can also be described in this way. In any case, such an equation will never result in the stock's value becoming negative.

7.3.2 Flow = variable * stock

The equation above is, in fact, a generalisation of the first two situations discussed, in that the coefficient is not a constant but a variable.

Example: waste load

The figure below shows how *waste load* is added and subsequently decomposes. In this example, the *decomposition time* is considered to be a variable, because it is assumed that the *decomposition time* will increase as the *total quantity* of waste load increases. The decomposition then equals *Total quantity / Decomposition time*. The coefficient in this case is *1 / Decomposition time*.

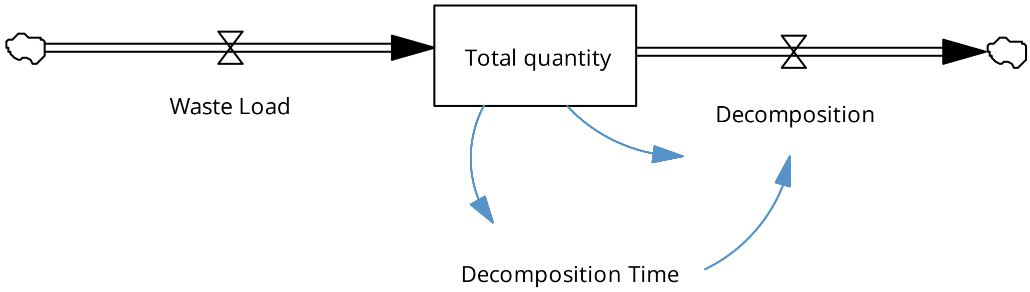


Figure 7.1. Decomposition time represented as a variable

7.3.3 Flow = (variable - stock) / constant

A well-known flow equation of this type is the classic negative feedback loop.

$$flow = (desiredvalue - stock)/adjustment\ time \qquad \text{Eq. 7.12}$$

A classic negative feedback loop has this general structure, in which the aim is for the value of the stock to achieve a desired level. The *adjustment time* relates to the speed with which the stock is adjusted. The flow can be both positive and negative.

Example: housing policy

The decision to build or demolish houses depends on the desired number of houses in a large city.

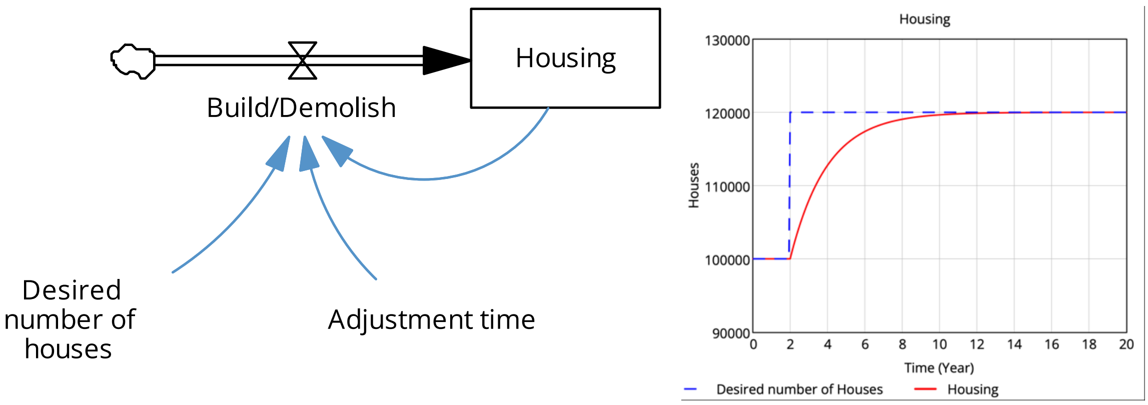


Figure 7.2. Housing as a function of time as a result of a step applied to the desired number of houses.

Figure 7.2 shows a Vensim representation of this situation. At time 0 the *desired number of houses* changes from 100000 to 120000 houses. The figure shows the response of the output variable (*housing*) to a step input variable (*desired number of houses*). The figure also shows the effect of the adjustment time of 3 months.

The model shows goal-seeking behaviour: the actual number of houses (Housing) moves towards the desired number of houses, i.e. the goal. A sudden decrease in the desired value would result in a similar response, except that the flow would be negative, decreasing the number of houses towards the lower goal.

Delays and smoothing are two specific applications of flows of the type of flow = (variable - stock)/constant and will be discussed in section 7.4.2.

7.3.4 Flow = normal flow + effect

A more complicated flow may be formulated as a normal or reference flow that is adjusted by adding one or more effects, or by multiplying the flow by one or more effects.

Example: housing portfolio

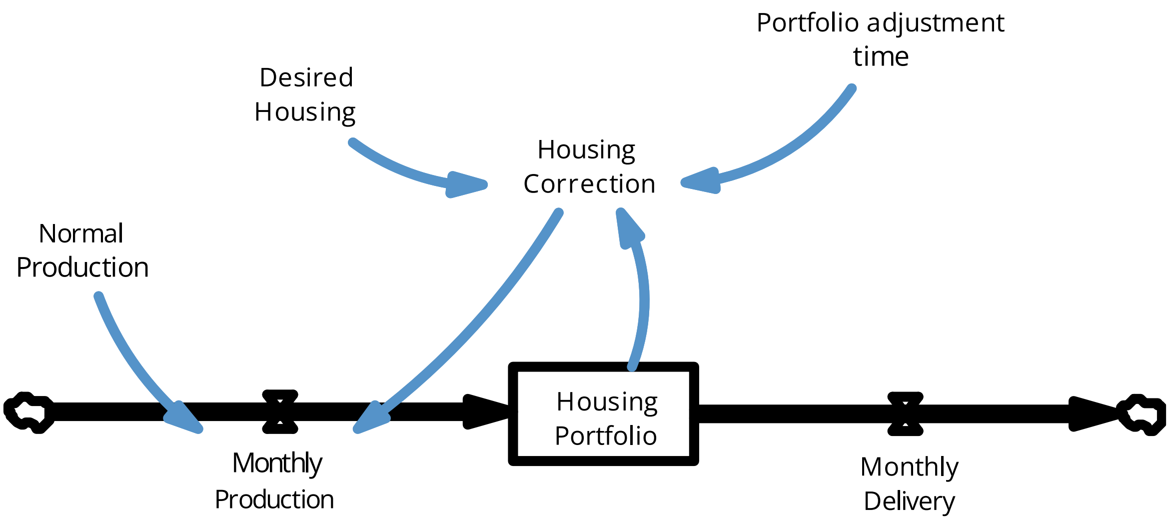


Figure 7.3. Housing portfolio

A large building company adjusts the normal production with a quantity of houses that ensures that the housing portfolio remains at the desired level. As a result, the production equation has the form normal flow + effect. If there are too many houses in the portfolio, the production of houses will be adjusted to below the normal flow. In such an additive equation, the units of all elements of the equation must be the same. Note that in such an equation, the monthly production could become negative if the housing portfolio is large enough. This cannot happen in reality, which indicates that an additive equation for a flow needs to be used with care.

7.3.5 Flow = normal flow * effect

Multiplication of effects, as shown above, is frequently used in SD models. The flow often consists of an average flow that is multiplied by one or more factors (acting independently and simultaneously) that ensure that the flow becomes larger or smaller.

Example: migration

Migration into and out of a city is influenced by the available housing in the city, amongst other aspects. Figure 7.4 shows that migration is determined by the housing fraction, that is, the number of houses in the city divided by the number of potential residents requiring housing.

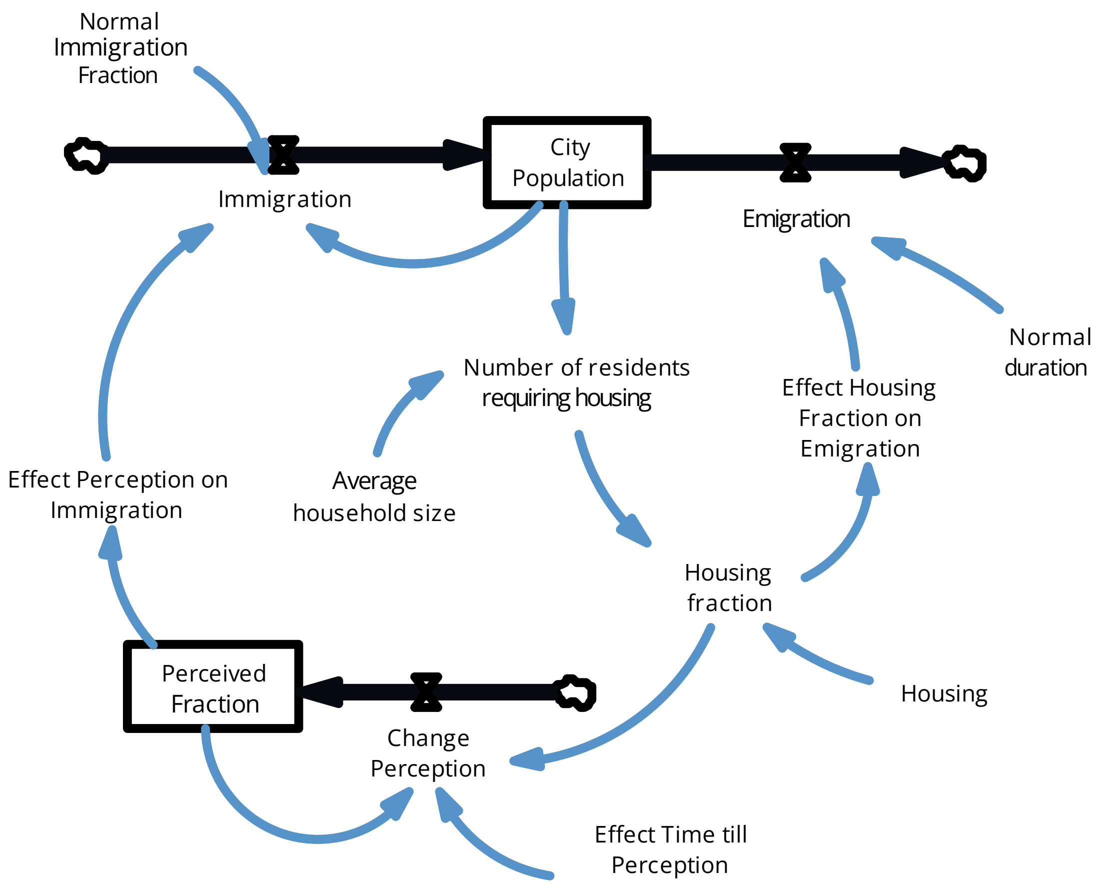


Figure 7.4. Population change as a result of housing availability in a city.

Immigration and emigration can be formulated in terms of normal immigration and normal emigration multiplied by a certain effect. In contrast to the additive effects discussed above, the multiplicative effects are dimensionless.

Although it may seem easier to aggregate input and output into a net flow, this can lead to a loss of insight. When multiplicative effects are used, we are dealing with positive and negative flows together and in it is very difficult to see whether and why the net flow is positive or negative. A strengthening effect may not increase the stock, as a positive effect and a negative normal flow will result in a negative flow. In such a case, the input and output may represent different processes that should not be combined.

There are differences between additive and multiplicative effects. A multiplicative effect can shut off a flow – in other words, set the value to 0. An additive effect generally does not shut off flows, unless a positive term in the equation happens to precisely equal a negative term. The effect is described as a changing proportion or percentage, or as the addition or subtraction of actual quantities. For example, is a car company considering a productivity increase of 10% per year or are they considering increasing the output by 100 cars/month? In general, multiplicative effects are used for percentage changes, while additive effects are used for absolute changes. Multiplicative effects are advised, particularly when more than one factor is acting upon the normal flow.

7.4 Auxiliaries

Every SD model can be built without auxiliary variables. Stocks are the "engine" of the model, where the integration takes place. Flows change the stocks. Besides some special functions which are hard to create with stocks and flows, you need nothing more. However, if you choose to create a model with only stocks and flows, you may lose the general overview. A relevant rule of thumb for avoiding over-complicating equations is "Richardson's rule" (as cited in Martinez-Moyano, 2012): combine a maximum of three different variables in one equation.

7.4.1 Parameters

Although there are only few parameters, such as real, physical elements that remain constant in the real world, there are many variables that can be assumed to remain approximately constant over a simulation run for a particular model and for a particular time horizon. Examples are conversion factors (e.g., productivity), reference values (e.g., the normal delivery delay), average lifetimes or residence times, adjustment times, et cetera. They can be included in a model as exogenous parameters if they are hardly, or not at all, influenced by other model variables.

Their values can be distilled from real data and existing knowledge about the relevant processes. However, they often are uncertain or inaccurate, as are models themselves. Hence, it is recommended to test the sensitivity of a model to small and larger parameter changes. The same is true for initial values. They are mostly inaccurate or uncertain and, especially in the case of highly non-linear models, require sensitivity and uncertainty analyses.

7.4.2 Delays

Delays occur in many systems. There are different ways to include delaying elements in your model; the choice for the method of representation depends on the system's characteristics. In this section we discuss two types of delays, material delays and information delays, as interconnections in systems can either be flows of physical material or flows of information. Delaying these two types of flows generates different behaviour. Additionally, the delays can be first-order or higher order. A special type of delay that is discussed is the pipeline delay. An overview of characteristics of material and information delays can be found in Table 7.1.

Table 7.1. Material and information delays

Material delay	Information delay
Used to slow flows of physical material	Used to slow a signal in information channels
Preserves material quantities	Preserves the range of input
1st order	1st order
2nd order	2nd order
3rd order	3rd order
… nth order	… nth order
Pipeline delay (infinite order)	

Material delays

A first-order material delay can be modelled in an SD model by a stock-flow structure of an inflow, stock and outflow equal to the stock divided by a time delay constant. The inflow is the signal to be delayed, and the outflow is the delayed signal. In such structures, the flows and stock are generally non-negative. Most SD software packages allow using functions to represent such a delay implicitly, for example DELAY1 in Vensim. The implicit DELAY1 function is mathematically identical to the explicit stock-flow structure of Figure 7.5, and thus generates identical behaviour.

If the outflow equals Stock divided by Time delay, then on average 1 / Time delay flows out per week.

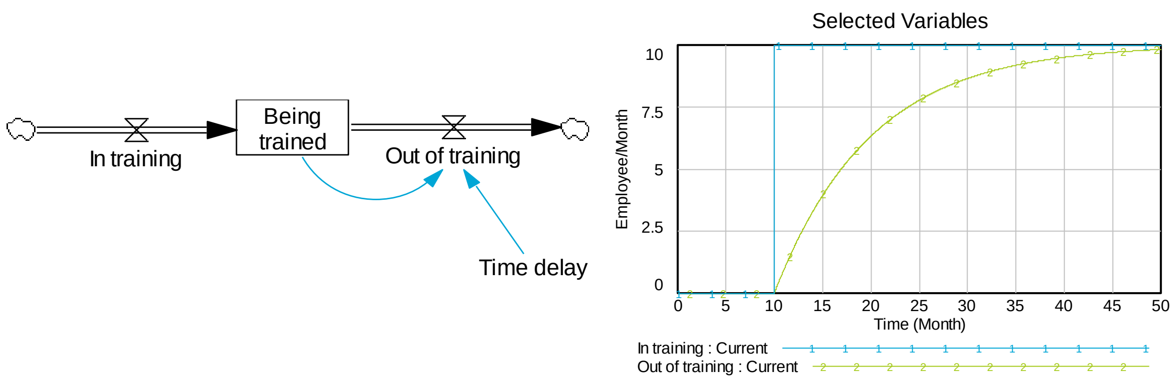


Figure 7.5. Left: Structure of a first-order material delay. Right: Behaviour of a first-order material delay, compared with its input (In training).

An example of a situation in which a first-order material delay is appropriate is a large company in which recently-hired production employees have to be trained.

Some will become fully productive in a short time, while others will take longer. Such a first-order delay is shown in Figure 7.5. The figure shows the response of the system to a step change in the input (i.e., *In training*). The output variable we are interested in here is *Out of training*.

One way to represent a first-order material delay is by specifying that the output equals the *Stock* / *Time delay*. We assume that the time delay in the example is 10 months. This means that each month, on average $1/10^{th}$ of employees in training finishes the training. Similarly, if the delay time equals 3 weeks, then on average $1/3^{rd}$ of the stock flows out per week. Note that the material in the stock (*'Being trained* in Figure 7.5) is assumed to be perfectly mixed and entirely homogeneous. The larger the time constant, the less steep the graph will be. Figure 7.5 shows that the output immediately responds to the input and moves towards a final value. A first-order delay is constructed using one stock variable, meaning that it is represented by a first-order differential equation. In contrast, a third-order delay is constructed using three stock variables in sequence, meaning that it is represented by a third-order differential equation.

Information delays

Information flows can also be delayed in SD models. Different model structures are used than for delaying material flows, because information, unlike physical material (matter), is not preserved. Information delays occur in complex systems as it takes some time for information about the system state to be perceived elsewhere in the system, or a variable that suddenly changes or oscillates needs to be smoothed. This is also called smoothing information.

For example, decisions in an organisation may be based on information from the past, in which the most recent information will usually weigh the most heavily. Smoothing can then be used to average out the weight of the information, thus taking more account of older information. Information can be smoothed using first-order information delays (also called *single exponential smoothing*). This structure is formulated in a net input flow (which can also become negative). The input flow equation formulated for this is:

$$netinputflow = (variable\ to\ be\ smoothed - stock)/smoothing\ time \qquad \text{Eq. 7.13}$$

The difference between the variable to be smoothed and the smoothed variable (stock) is divided by a smoothing time to represent the change in the stock (Figure 7.6).

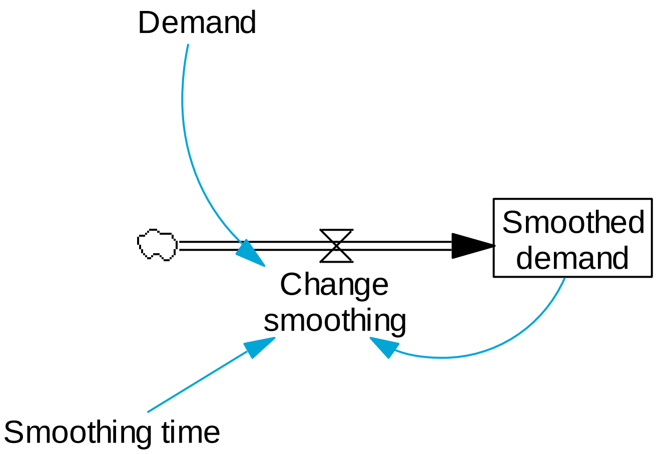


Figure 7.6. Structure of an information delay. Here, *'Demand* is the variable to be smoothed, *'Smoothed demand* is the smoothed variable, and the 'Change smoothing' input is defined as (Demand – Smoothed demand) / Smoothing time.

This structure can be used, for example, in situations where a product manufacturer looks at market demand, but does not respond immediately to every change in demand. In the case of smoothing, the manufacturer will not respond to the market demand, but to the smoothed market demand. This smoothed market demand will show fewer radical fluctuations. When a manufacturer responds to the smoothed input, smaller fluctuations will occur in the production than in the case of a direct response to the fluctuations in demand.

Figure 7.7 shows the effects of smoothing in a model in which the demand is the input and this input is smoothed using a smoothing time of 10 weeks. Noise has been added to the input (modelled by a random variable) to represent the effect of seasonal influences. For the purpose of comparison with the smoothing time of 10 weeks, the graph also shows the results for a smoothing time of 4 weeks. In both cases, the smoothed demand shows fluctuations with a smaller amplitude than the original demand. The figure also shows that a smoothing time of 4 weeks follows the input better, but gives larger fluctuations than a smoothing time of 10 weeks.

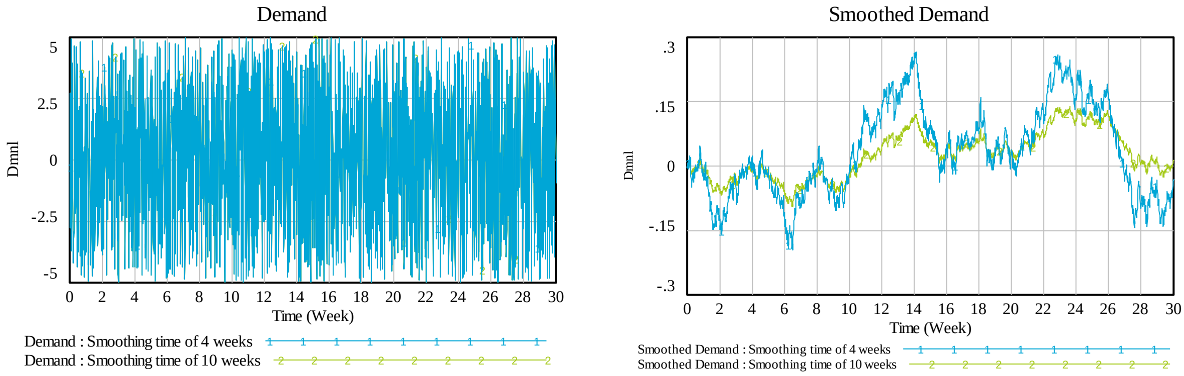


Figure 7.7. Effect of smoothing a randomly fluctuating variable.

As indicated above, a smoothed variable is determined at each time-step by taking the difference between the actual variable and the smoothed variable up to that point, and by dividing this by the smoothing time. This divided difference is then added to the smoothed variable up to that point to obtain the new smoothed variable. Consequently, the flow is the difference between the real and the smoothed variable divided by the smoothing time. This means that, as the smoothing time increases, the change in the smoothed variable (i.e., the flow) decreases.

Figure 7.7 shows the effect of smoothing for a varying input variable. A step function has been used for the demand to demonstrate what happens to the smoothed variable if a sudden change occurs in the input.

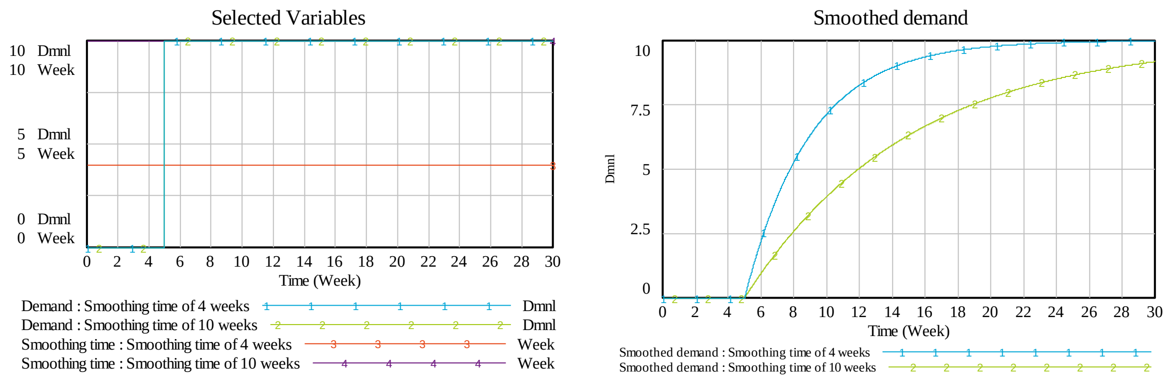


Figure 7.8. Effects of different smoothing times on the smoothed variable (right) under a step-wise change in the input variable (left).

Higher-order delays

In the previous section, first-order time delays were described. For some systems, however, a first-order delay cannot adequately represent the system behaviour. For example, in a situation in which new employees are first trained and then start to work, but it still takes some time before they are fully productive, two stocks can be used. The total training time of the employees is seen as two linked first-order stocks, which jointly result in a second-order delay.

Figure 7.9 shows four ways to represent a second-order delay. These result in the same value for the out variables if the time delay is constant.

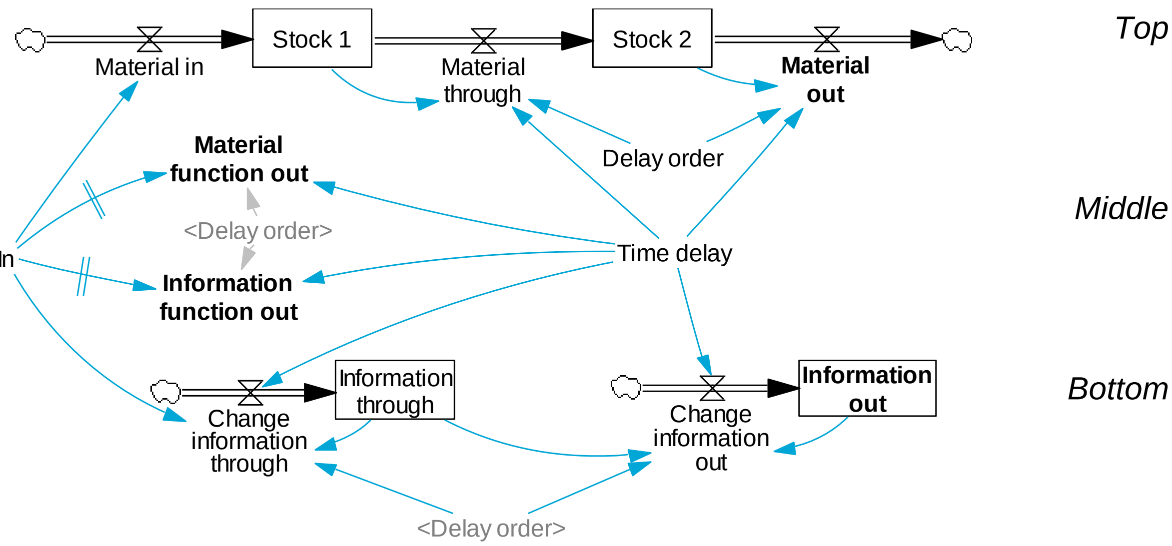


Figure 7.9. Four ways to represent a second-order delay (**bold**). The model has stock-flow representations of second-order material and information delays, in addition to the same delays specified in functions.

The Vensim equations of the four different delays are:

$$Material out = \frac{Stock2}{\left(\frac{Time\ delay}{Delay\ order}\right)} \qquad \text{Eq. 7.14}$$

$$Material function out = DELAYN(In, Time\ delay, In, Delay order) \qquad \text{Eq. 7.15}$$

$$Change information out = \frac{(Information through - Information out)}{\left(\frac{Time\ delay}{Delay\ order}\right)} \qquad \text{Eq. 7.16}$$

$$Information function out = SMOOTHN(In, Time\ delay, In, Delay\ order) \qquad \text{Eq. 7.17}$$

Here the *Time delay* included in the model and equations represents the total delay time. For consistency, between the delay functions in Vensim and the full model representation, the *Time delay* in the *Material out* and *Change information out* variables needs to be divided by the *Delay* order. This is important to ensure that the flows and stocks involved are only delayed by the time that applies to them individually and not by the total delay time.

The top of Figure 7.9 depicts a typical stock-flow structure for a material delay. The bottom structure is used to average information, but will result in the same value for the output variable. Note that the output variable in the top structure is a flow, whereas the output variable in the second construction is a stock. The middle two variables *Material function out* and *Information function out* contain functions for the same material and information delays. In Vensim, depending on whether a material delay or an information delay is involved, DELAY or SMOOTH are used (see Eq. 7.14 and Eq. 7.16).

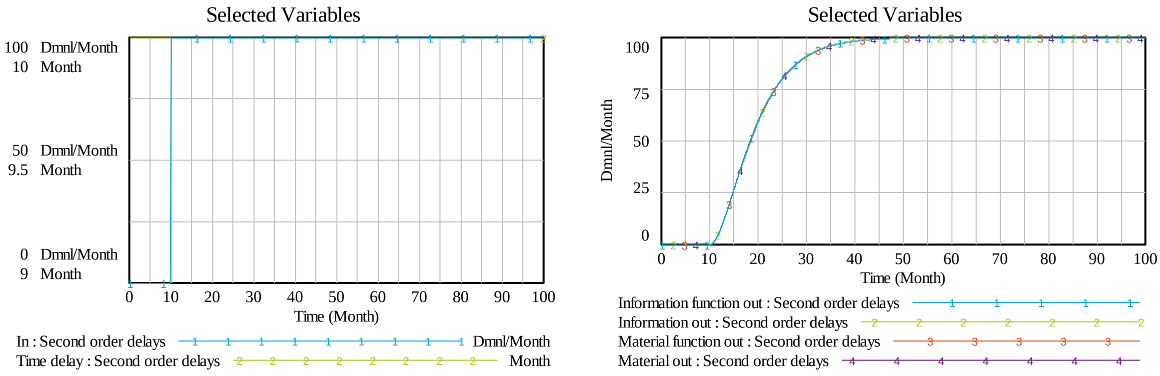


Figure 7.10. Results for the four representations of a second-order delay (*right*) from the same input and constant time delay (*left*).

Connecting three first-order systems results in a third-order delay. A third-order delay gives a somewhat more delayed response to a change in the input and will rise more sharply in the middle of the behavioural mode than a second-order delay (Figure 7.10).

Pipeline delays

A pipeline delay is by definition an infinite-order material delay. In a pipeline delay, the output variable precisely follows the input variable, except that there will be a time lag. A pipeline delay of 10 hours with a stepped input variable, therefore, results in a stepped output variable, except that the step will take place 10 hours later. In the example of Figure 7.5, a pipeline delay would mean that the modeller makes the implicit assumption that the training of every employee takes exactly 10 months, no more, no less. In practice, pipeline delays do not occur very often, and the response to a stepped input variable usually shows a less marked response.

In Vensim pipeline delays can be modelled with the function DELAY FIXED, in Stella with DELAY.

Comparing different orders of delays

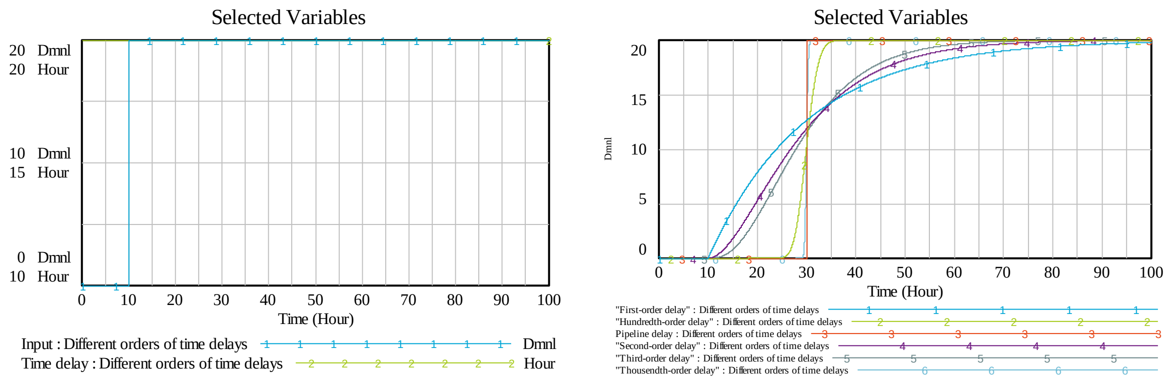


Figure 7.11. First-, second-, third-, hundredth- and thousandth-order delays and a delay fixed or pipeline delay.

Figure 7.11 shows how the different types of delays respond to a step function and constant time delay. The figure shows that increasing delay orders look more like a pipeline delay. The lower delay orders, particularly the first-order delay, respond most quickly to a changing input.

The choice of the order of the delay influences the model's behaviour, especially when dealing with steep input variables, such as a step function (in mathematics also called Heaviside function). Usually, the difference between a third-order delay and, for example, a sixth-order delay is not significant and falls within the uncertainty of a system. As a consequence, the choice of the delay order is in practice usually restricted to a first-order delay, third-order delay, or pipeline delay.

Comparing material and information delays

Information delays and material delays are building blocks to model the appropriate system behaviour. Material delays are delays of flows of physical material, with physical units. Information delays are delays in information channels, for example a change of perception, updated beliefs, or a way to smooth variables. If you compare system behaviour, the material and information flows only behave differently if the delay time is variable. Indeed, the differences between material and information delays only become clear when the time delay is not constant.

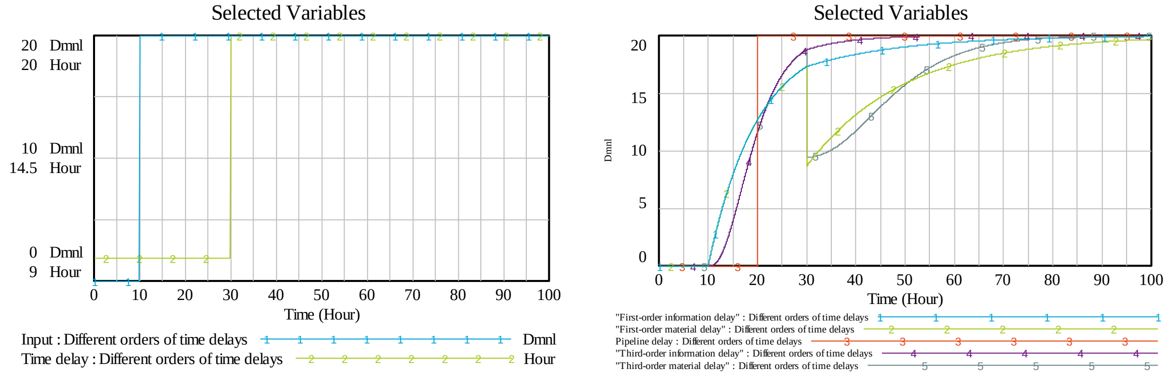


Figure 7.12. Typical responses of first- and third-order material and information delays, combined with a pipeline delay (*right*) of the same stepwise input and variable time delay (*left*).

Figure 7.12 shows the responses of first- and third-order material and information delays to a stepwise input and to an increase in the *Time delay* at t = 30 hours. Until the *Time delay* changes, there is no difference between the responses of the material and information delays to the stepwise input. However, the increase in *Time delay* is followed by a sharp decrease in the output of the first- and third-order material delays, but not the information delays.

This is logical, and follows from the model structures, but many still find it hard to understand. Consider two examples. First, a production firm needs tools for maintaining its machines, but have to deal with a delay in procuring these tools (e.g., two months). If the procurement process becomes faster (e.g., the delay becomes one month) and the number of orders remains constant, the procurement process will temporarily generate a higher output to reduce the stock of tools from the original two month's supply to the one month's supply that is now needed. To satisfy conservation of matter, and not simply lose materials from the procurement stock, it is necessary that this temporary rise in output occurs. The opposite happens if the time delay on procurement increases, with a temporary decrease in the output as a result. Second, consider someone who uses price information for trading stocks. You can imagine that not much would change if the running average generated by the information delay changes from two months to one month. In an information delay, there is no conservation of mass or energy, after all.

7.4.3 Time

Sometimes, you want to use *Time* as a variable in a model. *Time* is a predefined, independent variable in SD software packages. It can be used in representing variables that are dependent on time, such as sin and step functions or even lookups. Figure 7.13 shows the behaviour of this variable during a simulation run.

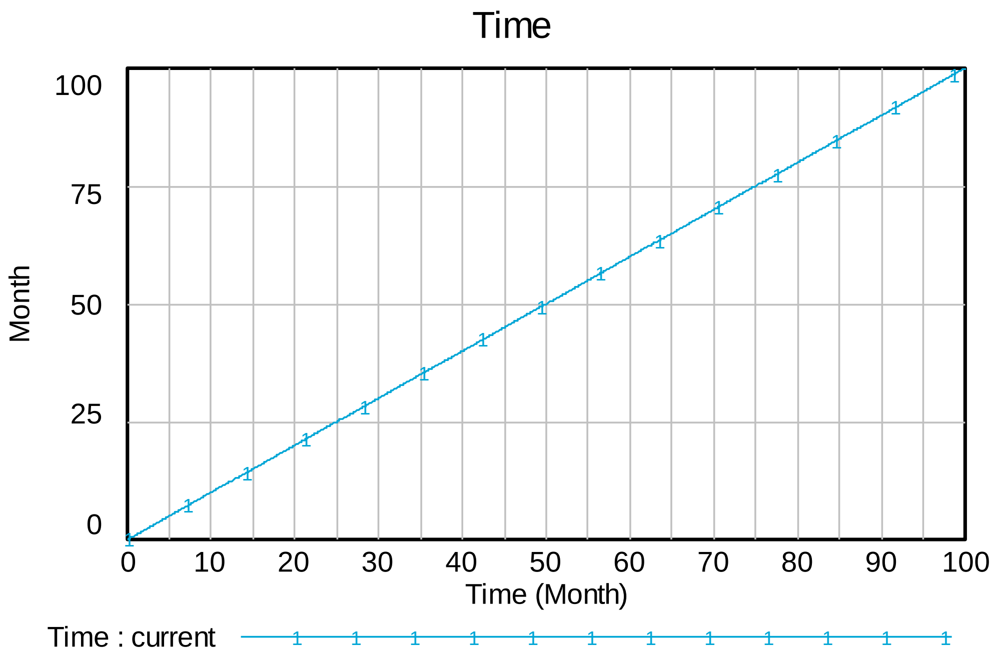


Figure 7.13. The value of "Time" during a run.

7.4.4 Table functions

Table functions or lookups are typical features of SD models. The term “table function” arose because of the sigmoidal form of many of the graphical functions used in such models. Table functions form a simple way to specify nonlinear relations between two variables, particularly when the relation between two variables is not easily represented in a mathematical equation. A table function determining the relation can be graphically formulated in the software intended for SD models (e.g., Vensim, Stella, Powersim). Alternatively, the modeller can formulate the auxiliary variable as a mathematical function that reproduces the desired behaviour. This may be difficult to formulate, but is easier to parametrise. The terms “graphical function” or “lookup” are synonyms for table functions. Lookups can also be used to introduce a time-dependent scenario into an SD model in Vensim.

Example: port development

A city has a new port development in which the number of terminals built each year depends on the fraction of land occupied. The effect that the fraction of land occupied has on the number of terminals being constructed can be represented in a table function. This table function has a multiplicative effect on the normal construction of terminals. This is shown in Figure 7.14. In Vensim, a table function can be formulated by selecting *Type* as ‘Lookup’ and subsequently clicking on the ‘as graph’ button on the right in the Equation Editor of the variable. The data points can then be entered numerically in the Input/Output columns or graphically by clicking in the diagram.

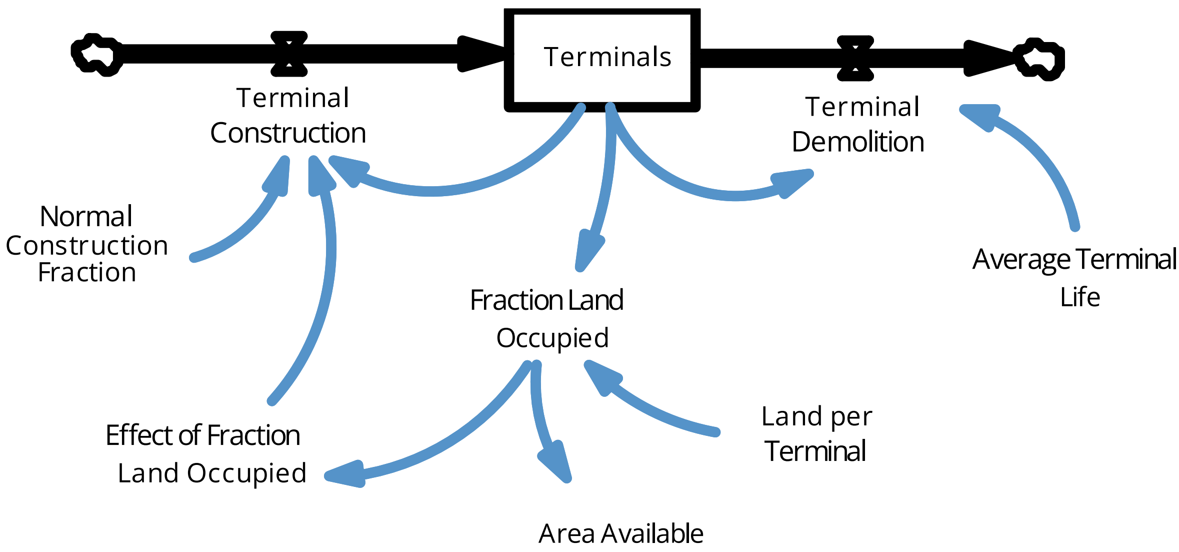


Figure 7.14. Terminals in a new port development.

The fraction of land occupied can in principle range between 0 and 1. The most important characteristic of the table function *Effect of Fraction Land Occupied* is that this must have the value 0 if the fraction of land occupied equals 1 (1 , 0) , because in that case no more terminals can be built there. Because a multiplicative effect is involved, the value of the fraction of land occupied for which the effect equals 1 must also be considered (i.e., the normal value applies in that case). The value of the fraction of land occupied for which the effect equals 1 (x , 1) implicitly determines the normal or reference condition of the site: a data point in the table that can serve as a benchmark against which system behaviour can be compared. We will define the reference condition as 60% of the fraction of land being occupied. This means that (0.6 , 1) is a point that should be defined in the table.

Following this, data must be found on the (normal) construction fraction. In this example we assume that this construction fraction equals 0.07 per year (7% new terminals per year) if the fraction of land occupied equals 0.6. This point in the table function is called the normal or *reference point*.

After specifying the reference point, it is important to consider the gradient and form of the function. At a certain moment in time, as land becomes scarcer, the more popular or easily accessible areas are already occupied and the remaining land becomes more difficult to build on. Here, the relation between the fraction of land occupied and establishment of terminals has a negative gradient. Figure 7.15 shows two alternative curves that conform to the conditions laid down so far.

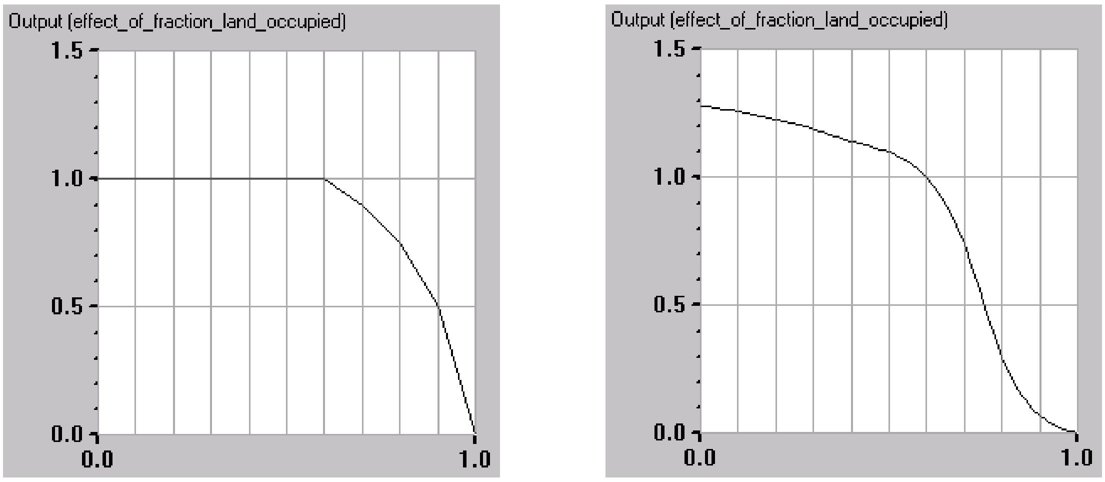


Figure 7.15. Two alternative curves (adapted from Richardson & Pugh, 1981). No further use allowed.

In the first alternative, the effect is 1 when the fraction of land occupied is less than 0.6. This means that at the start of development of the area, growth occurs with a constant factor (note that this constant growth factor is multiplied by the normal construction factor and the total number of existing terminals to determine the increase). As soon as the fraction of land occupied becomes larger than 0.6, the curve becomes steeper. The second alternative starts above 1 for low values of the fraction of land occupied up to the reference point. In this alternative, the gradient is the steepest for a fraction of land occupied of approximately 0.7. The alternative that corresponds best to the behaviour of the system and available data can then be selected.

We can also consider the possibility that the curve could first increase before decreasing, that is, it could have had a slight bell shape rather than being monotonically decreasing. Such a curve would have conformed to the conditions identified at the start. However, this curve tries to combine two different effects in one table, a positive and a negative effect. The positive effect might imply that as the new port puts infrastructure in place, more terminals will be constructed. Only later, when more land is occupied, does this initially positive effect no longer apply. There is nothing wrong with using such a curve. However, in the curves defined above, the effect of less and less land being available is already included and, in the part with values of 1 or above, the presence of infrastructure is assumed.

The following guidelines are important in formulating table functions:

- A table function has several elements, including gradient, shape and one or more specific points;
- Pay attention to the gradient and shape at both extremes and in the middle of the table. When a table function levels out, this represents a saturation effect. If a table function is steep, the effect of any change is stronger;

- Determine the coordinates of as many of the following points as possible: where the y-value equals 0 and 1, where the x-value equals 0 and 1, extreme x- and y-values;
- For a table function representing a multiplicative effect, the point where the y-value equals 1 is the normal point or reference point. For a table representing an additive effect, the point where the y-value equals 0 is the normal or reference point.

The table function values can be distilled from real data and from existing knowledge about the relevant processes. However, their values can be uncertain or inaccurate. It is recommended to test the sensitivity of the model to changes in the table function values. See Chapter 8 (section 8.2.2 for more information on sensitivity analyses.

Example: population scenario input

SD models frequently use dynamic inputs that are independent from the rest of the system model. In these cases it is generally better and easier to use a scenario input in a table function or lookup. This means that you make the lookup dependent on the time variable in your model (i.e., by definition “Time” in a Vensim model, which can be introduced as a shadow variable).

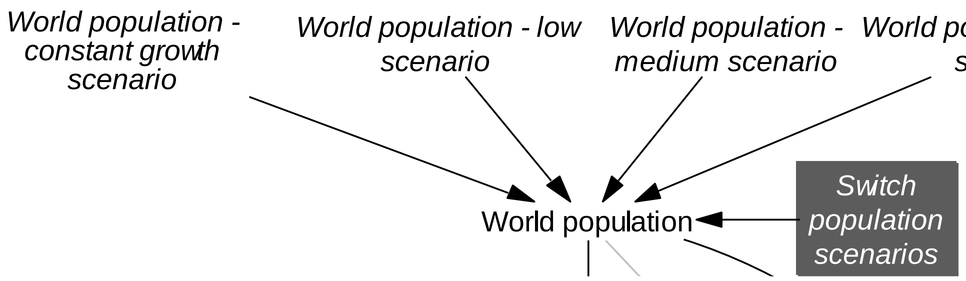


Figure 7.16. Part of a model with four different population scenarios as time-dependent inputs.

In this way, as shown in Figure 7.16, alternative scenarios can be introduced to the model, which allows both for linking the model to existing data, and for testing the model’s response to different input scenarios. The switch variable yields a constant value, with the *World population* variable using different inputs based on the constant value that the switch variable takes.

The Vensim equations of all time dependent lookups are either:

$$Auxiliary with lookup = \qquad \text{Eq. 7.18}$$

or:

$$Lookupvariable = [(starttime, minimumoutput) - (finaltime, maximumoutput)], (t_1, data_1), (t_2, data_2), (t_3, data_3), \dots, (t_n \quad \text{Eq. 7.19}$$

Here, the use of the *Lookup variable* has the syntax *Lookup variable (Time)* with the lookup input between parentheses. The only difference between the two types of lookups is that an Auxiliary with lookup can only be used with one input, while the Lookup variable can be used with different inputs in multiple locations in the model.

7.5 Standard structures

SD models frequently contain similar model structures. Three examples of these often-used structures are the procurement delay, the ageing chain and the co-flow. Of course, these standard structures can also be combined.

7.5.1 Procurement delay

The procurement delay represents the effect that, once you decide that you need to expand a capacity, it takes time before the new capacity becomes available. Examples of such delays can be found, for example, in models of resource extraction. Before the capacity of a mine can be expanded, the owners need to request permits, organise funding and order machines, for example. The procurement delay arose in the first simulation models made during the Second World War. Military planners found that they had to account for delays in arms procurement to assess accurately how much capacity would be available on the battlefield.

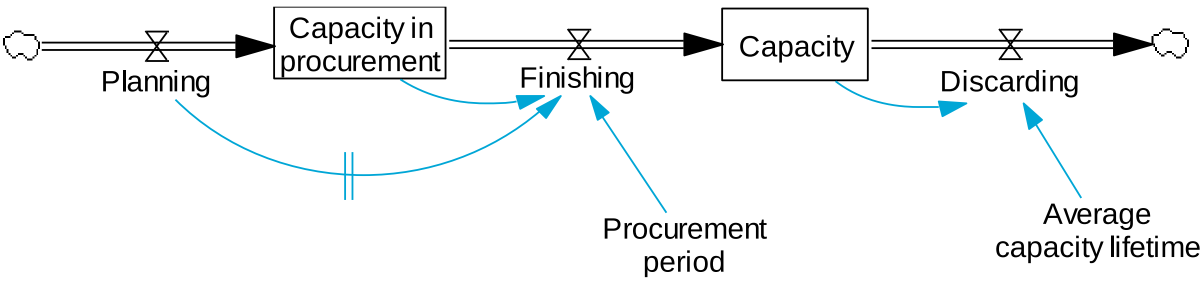


Figure 7.17. Typical procurement delay structure.

The structure of a procurement delay (Figure 7.17) generally contains two stocks and three flows: planning, finishing the preparation or procurement, and discarding the capacity at the end-of-life. The second flow can be a higher-order delay of the first flow, where the initial value of the delay function may be given by the stock divided by the procurement period:

$$Finishing = DELAY3I\left(Planning, Procurement\ period, \frac{Capacity\ in\ procurement}{Procurement\ period}\right) \quad \text{Eq. 7.20}$$

Discarding may be a standard first-order delay (i.e., *Capacity/Average capacity lifetime*), or can be a higher-order delay, depending on the nature of the discarding capacity.

7.5.2 Ageing chain

An ageing chain is in essence a set of interconnected, consecutive stocks. It is found in population models, but also for anything else that ages and where age is an important characteristic. Ageing chains thus allow us to model these systems in a less homogeneous manner than with a single stock.

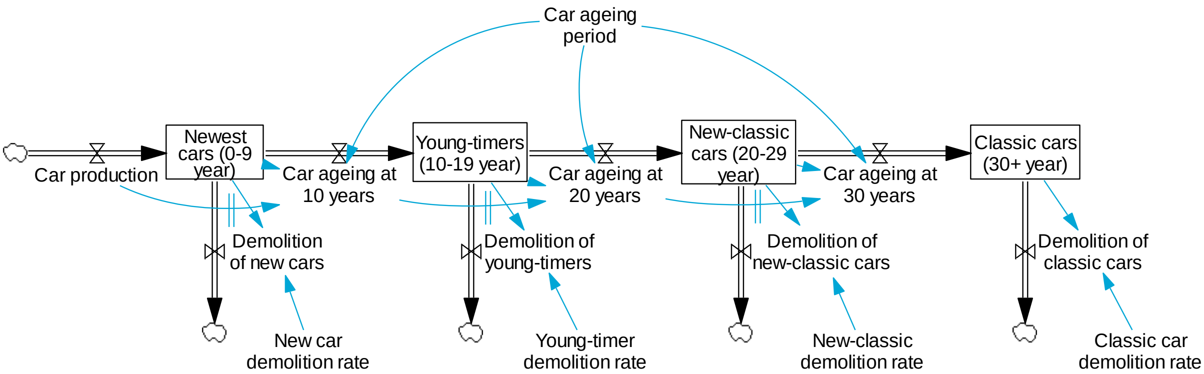


Figure 7.18. Ageing chain for cars.

Figure 7.18 shows an ageing chain for cars with a fixed ageing period per age cohort. It is also possible to vary ageing periods per cohort. As the demolition rates (or mortality for human or animal populations) are often not the same for each cohort, ageing chains allow for more accurate descriptions of population development. The accuracy usually increases with the number of cohorts represented in the ageing chain.

The flows in the ageing chain may make use of higher-order delays (e.g., third-order delays, similar to the delay function used in the procurement delay). The necessity for doing this decreases with the number of stocks in the ageing chain and with the continuity of the population development. If, in the example in Figure 7.18, the number of cars suddenly decreases or increases in one of the stocks (e.g., due to imports or demolition subsidies), it becomes more important to use higher-order delays. However, the use of pipeline delays is discouraged, as these only start reacting after the first ageing period, and so show less realistic behaviour.

7.5.3 Co-flow

The characteristics of a population in an ageing chain may be both dynamic and vary with the age. In such situations, it is useful to construct a co-flow next to the ageing chain. The co-flow should have the same structure as the original flow. The flows in the co-flow contain the development of the ageing chain characteristic, divided by the ageing period.

The co-flow contains the characteristics of an average individual from the corresponding ageing chain. Note that if the ageing chain has cohorts of non-equal size, the flows of the co-flow cohorts need to be disconnected.

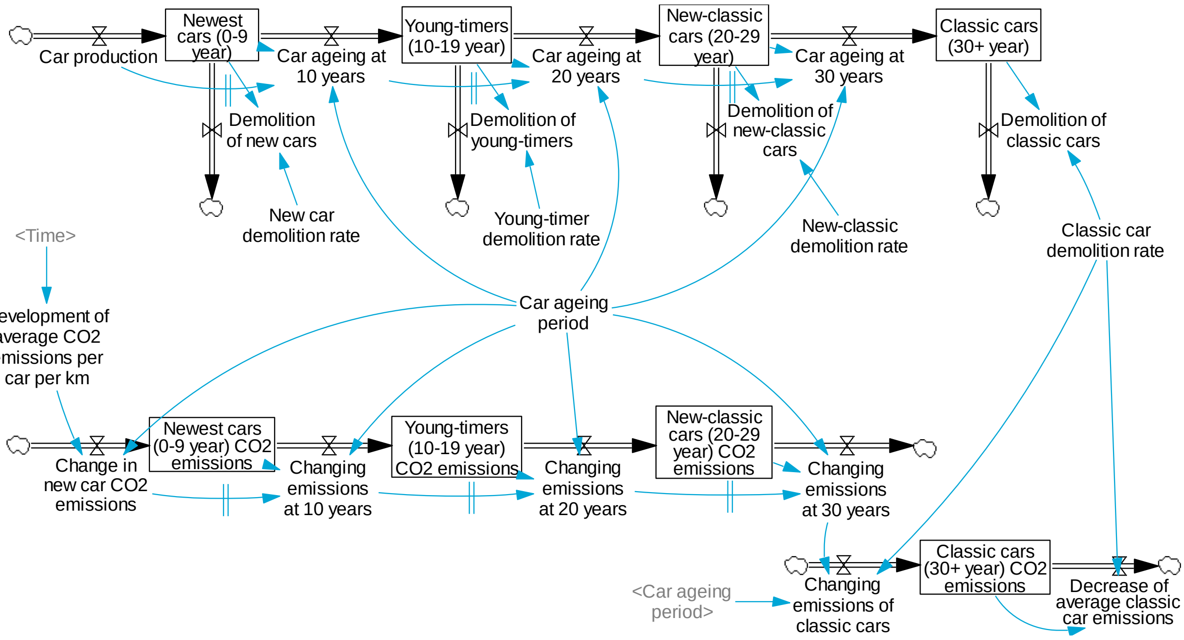


Figure 7.19. Co-flow structure connected to car population ageing chain.

An example of a co-flow structure is shown in Figure 7.19. In addition to the cohorts for the different ages of cars, there is another flow chain, similar in structure to the original ageing chain (i.e., with the same cohorts) – this is the co-flow. In the first inflow to the co-flow, the new CO_2 emissions per car per kilometre are divided by the car ageing period of ten years. As the “old” average flows out, the co-flow stock simultaneously and continuously keeps track of the average emissions per car of average age in the corresponding ageing chain stock. Ideally you should use the same order delays in the co-flow as in the ageing chain.

In the car example, the lifetime in the last stock (classic cars) is different from the lifetime in the other ageing chain stocks. If we assume the lifetime is higher (after all, cars over a century old still exist), you would get CO_2 emissions per car that are too high in this last cohort. You need to multiply the outflow of the *Changing emissions at 30 years* with the *Car ageing period*, and divide it by the lifetime in the Classic car co-flow, which in this case is 1/*Classic car demolition rate*.

Using this structure, you can calculate the total yearly CO_2 emissions of all cars, if you know how many kilometres cars of each age drive in a year. You therefore have to multiply the number of cars in each cohort with their corresponding co-flow stock, and the number of yearly kilometres for each cohort. If you then sum these yearly emissions per cohort, you will have the total yearly emissions from cars.

7.5.4 Switches

A switch is a structure which allows you to turn parts of a model on or off. Switches can be used, for example, to select input scenarios, switch between structural uncertainties, or turn policies on and off.

Switches contain three elements. First, the structure that needs to be switched on and off needs to be modelled. Second, the switch itself is modelled. This is a constant that should take specific values. Third, there needs to be a variable that contains an IF THEN ELSE structure to connect values in the switch constant to the respective structures.

Figure 7.16 shows how different input scenarios for the world's population can be selected. The *Switch population scenarios* is intended to have values of 1, 2, 3 or *ELSE*. The equation below shows how the IF THEN ELSE function is used in the variable *World population* to link these values to the four potential input scenarios.

$$\begin{gathered} Worldpopulation = \\ IFTHENELSE \\ \text{World population - low scenario}(Time), \\ IFTHENELSE \\ \text{World population - medium scenario}(Time), \\ IFTHENELSE \\ \text{World population - high scenario}(Time), \\ \text{World population - constant growth scenario}(Time) \end{gathered} \qquad \text{Eq. 7.21}$$

8. Evaluation

Model evaluation consists of several steps. The first steps, verification and validation, aim to check whether the model is suitable for the intended purpose. Verification and validation can also be considered to be part of model development and are therefore most often discussed in SD literature directly after the structure of the model. Verification checks whether the model is correctly coded. A check is performed to ensure that no errors have been made in representing the model in the computer. Another very important test is the dimensional consistency check. The right-hand side and left-hand side of every equation must match in terms of the dimensions. It must be borne in mind that in SD only parameters with a meaningful interpretation may be used. A check of the numerical integration method and step size also has to be carried out.

Conversely, the validation of a simulation model is about whether a model is fit for purpose (Oreskes et al., 1994) and builds confidence (Forrester & Senge, 1980). Investigating the suitability of a model for the intended objective is a very important part of a model study. SD often criticised because frequently informal, subjective or qualitative methods are used for model validation (e.g., a famous case was Nordhaus, 1973). Despite this criticism, several SD authors showcase more formal quantitative methods. This chapter discusses a wide range of both formal and less formal validation tests.

After verification and validation, it is important to interpret the model behaviour by linking the observed dynamics to the model structure. In this book, we use the various types of tests discussed by (Forrester & Senge, 1980). These make a distinction between tests of model structure and model behaviour. Other literature, most notably Barlas (1996) and Sterman (2000), makes a distinction between “direct structure tests” – tests of model structure without simulating the model –, “structure-oriented behaviour tests”, which test the model structure by simulating the model, and “behaviour reproduction tests”, which allow for statistical comparison of model output with past behaviour of the real system. Also, because of “Von Neumann’s elephant” (Mayer, Khairy, & Howard, 2010), being able to fit a large model (i.e., more than four parameters) to behaviour is not a relevant test of its validity.

Although a substantial part of formal model testing is conducted after the formulation of the model, in practice, tests will be conducted at every stage of the model cycle. This chapter will discuss the different types of tests that can be used for model validation. In validation, a distinction can be made between causal-descriptive models (*white box models*) and correlational models (*black box models*). In black box models (statistical models), we are concerned with the model’s output, and a model will be valid if the output, with the same input, corresponds to the real situation within a particular margin. This type of output validation is in fact a statistical testing problem. In white box models, the causal relations of the system are described. This not only involves the output, but the model's internal structure, as well. The model must in fact also contain an explanation of how the system behaves.

SD models are white box models and are, amongst other things, intended to study alternatives. This is only possible if the internal structure of the model contains the aspects of the system that are relevant to the problem behaviour. Therefore, a SD model must produce the right output behaviour for the right reasons, and this should be investigated during validation.

8.1 Verification

The verification of a model investigates whether the model has been coded correctly and consistently. In contrast to validation, the relevant tests do not require any comparison between the model and the real system. They are aimed solely at the question of whether the model has made the correct transition from concept to specification. This stage of testing can have significant overlap with the specification of the model, since the analyst will most likely test while building and iterate between testing and developing.

Three different types of tests will be described:

- Correct coding of the model;
- Dimension analysis;
- Numerical errors.

This is in no way a complete list of all available tests. The tests presented here provide guidelines, a small set of consistency checks that are the absolute minimum of tests performed on a model.

Once the errors in the model have been identified, the last part of the verification stage consists of debugging the model to make it more accurate.

8.1.1 Correct coding of the model

The first strategy to ensure the correct coding of a model is not a part of model testing but of building; prevention is less resource intensive than firefighting. Building a readable model will make it easier for the analyst to check his own work and forces a well-considered structure of the model. The goal here is to try to make the model easily understandable either to someone who joins the model development team later in the process of building the model, or a client who is going to use the model.

Aside from reading the individual equations and evaluating them on a case-by-case basis, one way to test whether the model has been coded correctly is to isolate sections of the model. As a first step, the modeller takes a section from the model for which the intended outcomes are known under known input. Then, (s)he tests whether these outcomes are actually generated by the isolated (separate from the rest of the model) section of the model under controlled input. This check is only to verify whether the isolated section performs as intended by the modeller, i.e., whether it is coded correctly. Whether this intended behaviour is suited for the purpose of the model is a question left for the validation.

8.1.2 Dimension analysis

In the dimension analysis the modeller performs two checks. First of all, the modeller checks whether the units of the model correspond with what the variables represent in the real world. For instance, in a model of a car factory, production capacity could be modelled as car per time unit (car / month, if the time unit of the model is month), i.e., the number of cars the factory can produce over a certain time interval. If that capacity is modelled as a stock, the rate of change of that capacity should be car / month^2.

The second check should be whether these dimensions have been coded correctly. Some software automatically checks whether the output unit of a variable corresponds with the incoming variables and the equation of the variable; if these do not match, the software reports an error. However, due to several reasons this check is not sufficient:

- The checks the software makes can be circumvented by “clever” modellers. For instance, in Vensim, the modeller can influence the outcome unit of an equation by multiplying it with the number 1 of a certain dimension, a so called “corrective constant”.
- The software can never perform the semantic check whether each variable has the correct unit for what it should represent.
- Certain functions can be very flexible in what kind of units they have as their outcome. For instance, in the GRAPH function in Vensim, the modeller can manually set the unit of the output of the function.
- Not all modelling software provides this feature. Therefore, the modeller should still be able to check should be whether the dimensions have been coded correctly without the help of software.

8.1.3 Numerical errors

Two different classes of numerical errors can occur when numerically solving a model: numerical method-dependent errors and model-dependent errors.

Numerical method-dependent errors

This class of errors arises from the selection of a numerical solver and/or time step that is unsuited for the developed model. They can be fixed by selecting a different method or adjusting the current method’s settings.

The incorrect choice of method emerges when the behaviour of the model is dependent on the choice of integration method. This can be detected by comparing runs using one method with runs using another. It is up to the modeller to determine which outcome is correct. A classic example is a model that should display stable oscillation, but displays expanding oscillation when Euler is used (Figure 8.1). For a more extensive discussion on numerical methods see Boyce and DiPrima (2005, CH 8), Borelli and Coleman (2004, pp. 122-129), or Strang and Herman (2016, theorem 4.1 and following).

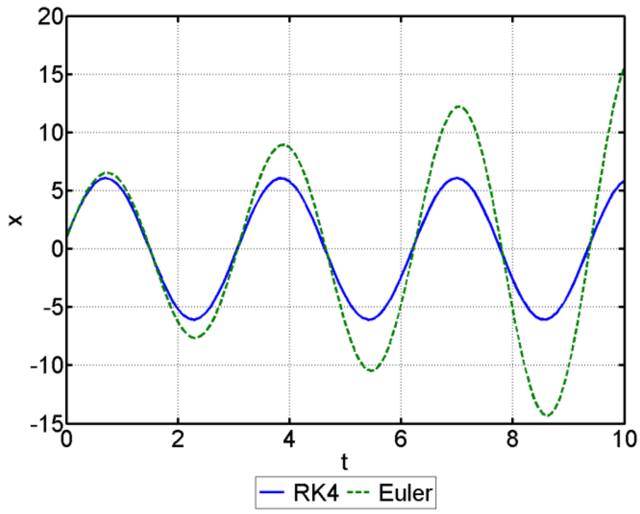


Figure 8.1. Results of an incorrect choice of numerical method. The model should give stable oscillation, as per the RK4 method. Using Euler, however, results in an increasing amplitude of the oscillation.

The second type of error is an incorrect choice of time step, where a reduction in the step size results in a significant change in the behaviour of the model (Figure 6.2). The solution is to reduce the step size to a size at which a reduction in step size no longer results in a significant change in behaviour.

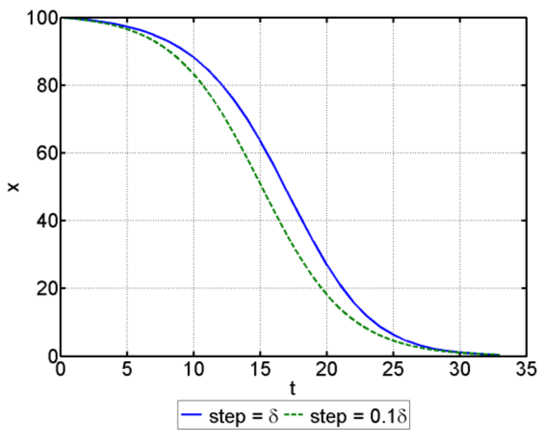


Figure 8.2. Incorrectly chosen step size. Note that, while there is a significant difference in behaviour for the two step sizes (δ), the end values are practically the same.

To check for these types of errors, the model can be run with different methods and settings for the time step. The most basic version of this tests checks for the results of different methods of integration and compares the original run to a run with the time step halved.

Model-dependent errors

The model dependent errors are a result of incorrect formulation of the model. Their magnitude and behaviour can depend on integration method and step size, but they originate from within the model, not the integration method. For instance, an equation causes erratic behaviour where behaviour should be continuous, or close to continuous. The solution of this type of error will always lie in reformulating that section of the model that is responsible for the erroneous behaviour.

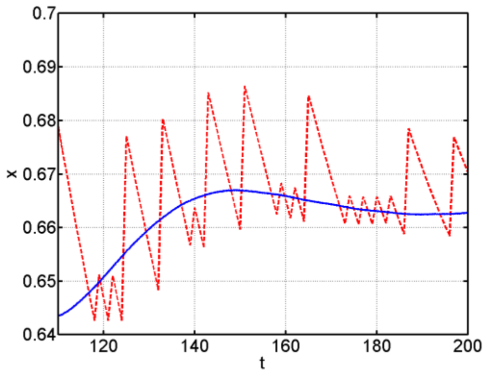


Figure 8.3. Typical behaviour of a numerical error that is dependent on the time step of the numerical solver. The model should display smooth behaviour, as per the solid line, but instead gives the non-solid line.

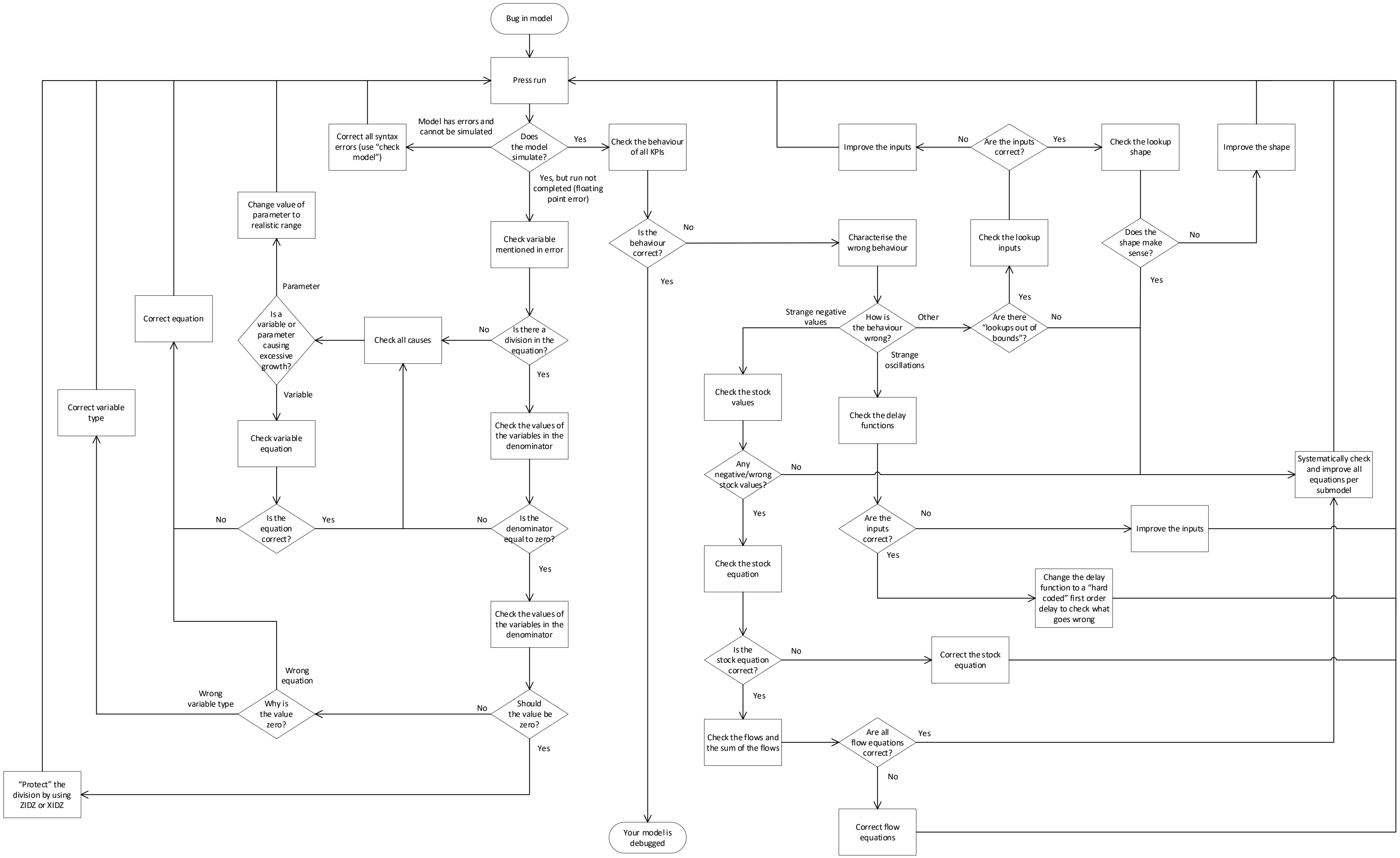


Figure 8.4. Flowchart for the debugging process of SD models (Auping & d'Hont, 2023)

8.1.4 Debugging

All modellers make mistakes. Some of these mistakes are clearly wrong and need to be addressed by debugging the model. Debugging is removing errors (so-called bugs), both in syntax and in logic, in your model. The process of debugging can be helped by correct coding, dimension analysis and avoiding numerical values, but often goes far beyond that. There are some rules of thumb on where to start, but it is important to realise that debugging only gets better with a lot of practice: debugging is built mainly on tacit knowledge and thorough understanding of SD models.

Novice modellers often make specific mistakes. We tend to see the following mistakes with novice modellers (please note that the list is not intended to be exhaustive):

- Wrong syntax in equations, especially when using functions;
- Wrongly used delay functions, especially using material delays from stocks to flows in procurement-delay structures. These cause oscillations due to an added balancing feedback;
- Summation of stocks in another stock; for example, summing the various cohorts of a population in another stock variable;
- Divisions by zero: division of one variable by another variable which may become zero. This also happens together with the previous mistake, if the initial value of the stock summing the other stocks is chosen to be zero;
- Mistakes with placing brackets, for example if a variable needs to be multiplied with one minus a certain percentage. If the brackets are forgotten, this may cause negative values;
- Wrong operators in stock equations; sometimes people feel the urge to correct operators like minuses and plusses in stock equations, which may lead to mixing them up;
- Wrong lookup or table functions; in lookups and table functions, all kinds of mistakes can be made. The wrong input can be used, or the order of inputs and outputs can be switched, leading outputs to be inputs and vice versa.

Based on these observations, we made a debugging flowchart to help students make their debugging process more efficient. Figure 8.459 presents the debugging flow chart resulting from our analysis. It contains roughly seven different “debugging loops” (some loops overlap partially, making an exact count impossible). The loops are related to: 1) removing syntax errors, 2) fixing floating point errors due to excessive growth, 3) fixing floating point errors due to divisions by zero, 4) fixing negative stock values, 5) fixing incorrect delays, 6) fixing incorrect lookups, and 7) the nuclear option of systematically checking and improving all equations per sub-model. It should be clear that option 7 should be avoided at all costs.

In debugging, the centre piece of the exercise is pressing "run". If this fails, because the model contains syntax errors, it is crucial to remove these first. Error messages in SD modelling software (please note that it is actually relevant to read these!) help to point out which variables are concerned.

The next step is pressing "run" again. If the model still fails, the second and third debugging loops focus on floating point errors which cause model runs to be not completed. To the best of our knowledge and experience, floating point errors are the sole reason of uncompleted runs, and floating point errors can only have two causes. The second debugging loop aims to fix excessive growth (often due to parameter values near inflows that are too high). The third loop aims to correct floating point errors due to a division by zero. These can be caused by a division through a sum of stocks which is wrongly modelled. For example, if the variable type of the *total population* in the COVID-19 model is not auxiliary but level with an initial value of 0, any division by the total population will cause a floating point error. The division by zero may, however, also be caused by any other denominator becoming zero, whether that should be possible or not.

At some moment, the error messages you receive after pressing "run" will not help you anymore. At that point, debugging will focus even more on behaviour of performance indicators in the model or other state variables. In debugging loop four, the aim is to solve wrong negative values which can be caused by wrong stock values, which in turn originate in either wrong (altered) stock equations, or incorrect flow equations.

In debugging loop five, the modeller should correct strange oscillations in behaviour that may be caused by incorrectly formulated (material) delay functions. This is a frequently made error, even by intermediate modellers. The basic error is conceptual in nature and concerns the fact that in the delay function not a conceptually similar variable is being delayed (e.g., a flow), but a conceptually different variable (e.g., a stock). Especially in the case of delay functions in stock flow structures, this error causes the introduction of an additional balancing feedback loop. The consequence of this additional balancing loop is seemingly unexplainable oscillatory behaviour. Changing the input of the delay to the stock's inflow solves the issue.

In the sixth debugging loop, the modeller tries to find incorrect lookup functions. The easiest mistake is a wrong input, for example, TIME STEP (which is obviously static) instead of TIME if a time input was needed for an input scenario. Other mistakes, which are sometimes more difficult to find, originate in switching input and output values in the lookup definition.

In some situations, the previous six debugging loops are insufficient to find the error. In these cases, modellers need to resort to systematically checking and improving of all equations per sub-model. Unfortunately, this is sometimes necessary, but it is a laborious and tedious task which is better avoided.

Note that in programming it is customary to first look at new pieces of the model when debugging. This may work in SD, but not always. For that reason, this step is not included in the debugging flowchart. This is because the feedbacks in the model may cause an earlier bug only to occur after a new part of the model is added. The bug is than not in the new part, but in an already existing part.

Floating point errors

Floating point errors are the only non-syntax errors, to the best of our knowledge, that can cause a run to be not completed. As such, they are the most important bugs to eliminate. Floating point errors can have two causes:

1. Division by zero;
2. Extreme exponential growth.

A division by zero is easily found, as the error will point out the equation in which the division takes place. The error is that the denominator is equal to zero, which is generally caused by one variable which has value zero, but should not. You need to check whether that variable has the right equation or is of the right type. For example, if a variable is a stock with initial value 0, but should not be a stock, you have to change the type to auxiliary. Otherwise, you change the equation. Only if the value should be allowed to be zero, you can protect your model by using a ZIDZ (i.e., Zero If Divided by Zero) or XIDZ (X If Divided by Zero) function.

Extreme exponential growth is more difficult to find, as the cause of the exponential growth is generally a feedback with the variable that gives the error. Therefore, you need to backtrack all causes of the variable mentioned in the error until you find a parameter with a wrong value or a wrong equation, and correct it.

8.2 Validation

Forrester and Senge (1980) distinguish two types of tests: tests of model structure and tests of model behaviour. This chapter discusses these tests and gives examples of how to perform these tests. We made some alterations to their paper, as one of the tests (extreme-conditions test) is considered by Forrester and Senge to be a test of model structure, while it relies on simulation of the model, and is thus a test of model behaviour. The dimensional-consistency test of Forrester and Senge is considered to be part of verification in this book. Finally, the extreme-policy test is in fact a form of the extreme-condition test.

Many of these tests can also be supported by experts or stakeholders. For example, boundary-adequacy tests of both structure and behaviour, but also structure-verification tests, behaviour-reproduction and surprise-behaviour tests can be performed by presenting the model structure during model formulation and evaluation to groups in sessions of joint sense-making.

When performing validation tests, it is useful to follow and report the following steps:

1. Which test are you performing, and what type of test is it?
2. How do you perform this test?
3. What do you observe?
4. What do you conclude: is your model fit for purpose?

8.2.1 Tests of model structure

Although all validation tests of an SD model are aimed at establishing confidence in the structure of the model, the tests in this section directly assess structure and parameters, although without investigation the relationship between structure and behaviour.

Structure-verification test

In this test you compare the system structure with the real-world system. Simply put, the model structure must not contradict knowledge about the structure of the real system. Typically, the *structure-verification test* is first performed with the modeller's personal knowledge and then extended to others who have direct experience with the real system, such as experts, stakeholders and other actors (we also call the process of engaging other people to evaluate model elements '*face validation*'). Do they recognise the positive and negative feedback loops? Does the model structure match the observable goals, pressures and constraints of real decision makers?

Verifying that the model structure exists in the real system is a relatively easy test that requires not as much skill as other tests. Many structures can pass the structure verification test.

Parameter-verification test

Similarly to the structure-verification test, the parameters must also correspond to the relevant descriptive, observed knowledge of the real-world system. Does the parameter conceptually correspond to elements of the real system structure? For example, *conceptual parameter verification* of a parameter such as 'normal number of contacts per week ' (Table 8.1) would involve examining determining how many face-to-face contacts the average person experiences per week. The parameters must also be checked on the basis of any numerical knowledge. For example, *numerical parameter verification* would determine if the value given to the parameter falls within a plausible range of values.[3]

Boundary-adequacy (structure) test

Whether the model has appropriate boundaries is a recurring question, as the adequacy of model boundaries can be verified structurally and behaviourally. Does

---

3 Graham (1979) discusses a variety of techniques for parameter estimation in system dynamics.

the model contain the structural relationships necessary to satisfy the model's purpose (i.e., is the model *fit for purpose*?)? Which variables are modelled endogenously and exogenously, and why? Is the model appropriately aggregated?

The modeller must determine that the model contains the most important concepts required to analyse the problem. Given the model's stated purpose, the boundary must be sufficiently broad, so that the model contains an endogenous representation of the most important mechanisms, but if the model is too expansive, it will no longer be transparent and less balanced.

For example, say that we have a model built with the purpose of understanding the growth of the tourist industry in a certain region. This model does not include any information about the wildlife in the region. This is only a valid boundary if the wildlife in the region has no influence on the tourist industry and if the wildlife is not considered to have any value on its own (e.g., if no legally protected wildlife species live in the area). If the recreational activity in the region influences the wildlife and the wildlife forms a significant part of the attractiveness of the region, the model should include some representation of the wildlife. Therefore, the inclusion of a wildlife sub-model is completely dependent on the previously specified model purpose or hypothesis.

Note that criticism of the model boundaries (e.g., "wildlife should be included in the model") is sometimes actually criticism on the model's purpose ("we actually want a model that focusses on wildlife, instead of a model for tourism").

Extreme conditions test

Another structure test is the *direct extreme conditions test*, in which the model equations are evaluated under extreme conditions. This test is done by determining whether the resulting values are plausible when compared to the knowledge or expectations of what would happen in the real situation. It is often easier to imagine what could happen in reality under extreme conditions than what could happen in reality without extreme conditions. For example, if the population is set to 0, there can be no births and there will be no consumption; if there is enormous pollution in a city, the death rate must go up and migration must go down; if there is extreme rainfall, floods occur. Every model equation can be checked this way by entering extreme values for the input variables and by comparing the output value to the expectations in the real situation. No model run is conducted for this; every equation is viewed separately, which is why this test belongs to the direct structure tests.

A model should be questioned if the extreme conditions test is not met. It is not an acceptable counterargument to state that these particular extreme conditions do not occur in real life and therefore should not occur in the model, because the nonlinearities introduced by approaches to extreme conditions can have important effects in normal operating ranges. The extreme conditions test is a powerful test for discovering flaws and for identifying nonlinearities and asymptotes that should be included in the model structure. For example, the extreme condition of a very tight labour market would affect production negatively and capital investments positively, so these relations should be included in the model. The extreme conditions test is a strong test that can aid in determining under which conditions the model works, and inform scenario building and policy analysis for policies that force a system outside historic behaviour.

8.2.2 Tests of model behaviour

Tests of model behaviour evaluate the model structure indirectly, by analysing the behaviour generated by the structure. Documenting hypothesised model behaviour is important. If a comparison is made with anticipated model behaviour instead of with information from the real system, the modeller must first record the anticipated model behaviour and, if necessary, graph this. Then the hypothesis can be checked: does the model behaviour correspond to the modeller's expectations? Additionally, any unexpected behaviour can be analysed to see why it occurred. If the model shows unexpected behaviour, two conclusions can be drawn: either the model is incorrect and must be adjusted, or behaviour has been found that might also occur in the real system under similar circumstances.

Sensitivity analysis (a.k.a. behaviour sensitivity test)

A sensitivity analysis is designed to determine the elements in the model to which the model is sensitive, and that have a major influence on the behaviour when they are changed. Sensitivity analyses are essential, both in model validation and in model use. In model validation, the goal of a sensitivity analysis is to determine how sensitive the model is to plausible changes in data. In contrast, during model use, a sensitivity analysis is aimed at locating changes in the system that have the desired influence on the system's behaviour and in this way contribute to the solution of a problem.

Richardson and Pugh (1981) mention three types of model sensitivity (i.e., *not* three types of analysis, but three types of results):

- *Numerical sensitivity.* Numerical sensitivity exists if a change in the assumptions changes the numerical value of the results. All models show numerical sensitivity.
- *Behavioural sensitivity.* Behavioural sensitivity exists if a change in the assumptions changes the pattern of behaviour of the model. For example, there is behaviour sensitivity if different assumptions change the behaviour from a gradual adjustment to oscillating behaviour. Behavioural sensitivity may be a negative indication for the suitability of a model or may indicate that the sensitive assumption can be used for intervention.
- *Policy sensitivity.* Policy sensitivity exists if a change in assumptions reverses the influence of a proposed policy. An example is if a reduction in the national

debt would result in an improvement of the economy, whereas in case of other assumptions the economy would decline.

A sensitivity analysis is used to study whether the model is sensitive to plausible variations in the parameter values and to plausible alternative model structures. Confidence in the model is enhanced if such alternative parameter values are not found. An important aspect is also whether the conclusions of the model study will change when the parameter values or the structure undergo a plausible change, because in that case drawing these conclusions is no longer justified.

The entire model is to be tested by increasing or decreasing each parameter a little bit (e.g., by 10%) and observing the effect of that change. In studying the parameters, the initial values of the levels and the table functions are also involved, as these are in fact made up of a series of parameters or approximations of polynomials of which the coefficients are parameters. It is also important to simultaneously study the effects of changes in different parameters. The modeller must not only check whether end values are changing, but also if the shape of graphs (trends) changes.

If is it shown that the model is sensitive to a particular parameter, this may imply that the parameter must be determined accurately if the degree of sensitivity is important in view of the objective of the model study. If the model has already been evaluated and suffices given the objective, sensitivity may be an important leverage point for a policy.

Sensitivity analysis

In a model of the early phases of the COVID-19 epidemic in the Netherlands (see Exercise 5.8), we changed all uncertain parameters by +/- 10% to do a sensitivity analysis. In Figure 8.5, it is shown how this model behaves over 200 runs. To generate this set of runs, the simulation software created 200 experiments with values sampled between the minimum and maximum values for the listed parameters. We use selected Latin Hypercube sampling to generate these samples (McKay, Beckman, & Conover, 1979).

We observe that for all runs, the behaviour is similar, although there are numerical differences. This is caused by the fact that model behaviour is largely “driven” by a lookup mimicking social distancing measures in the model. As a consequence, the infection rate causes the peaks in the infectious population, and the related actual reproduction number is not influenced sufficiently to flip around 1 (i.e., if the reproduction number is over 1, the number of cases increases, and if it is under 1, it decreases). Therefore, the model is numerically sensitive to the inputs and not behaviourally sensitive.

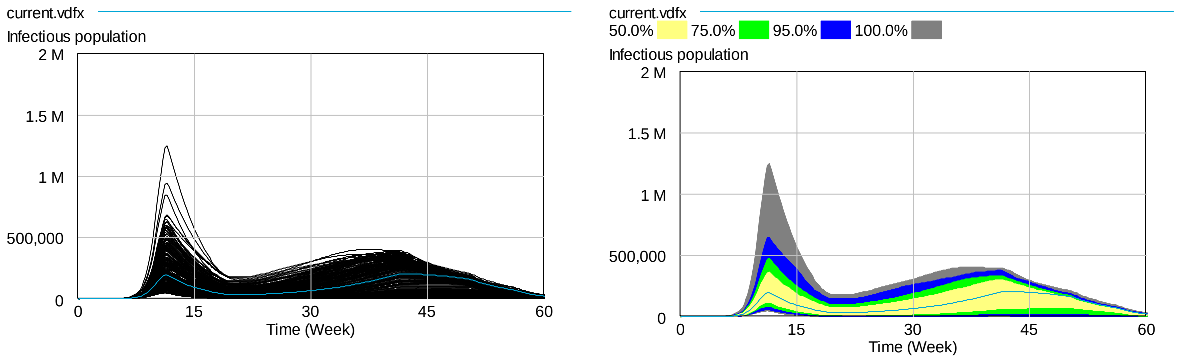


Figure 8.5. Two ways of showing the results of a sensitivity analysis.

Behaviour-reproduction test

Behaviour-reproduction tests examine how well the model-generated output matches the observed behaviour of the real system – especially the behaviour characteristics of the system that are important for the analysis of the problem. We briefly discuss four types of tests that focus on behavioural characteristics: trends, modes, frequencies, phases, and amplitudes. Consider these tests approaches to examine whether or not a model generates correct patterns of behaviour. To perform such tests in practice, a model may need to be excited by a simple test input, depending on the nature of the behaviour. For example, random disturbances are useful for systems that exhibit damped oscillation.

- A *symptom-generation test* investigates whether the simulation model recreates the symptoms of the problem that was originally formulated. Thus, if the modeller cannot show how a specific undesirable situation arises through policies and structure, the model is of no use for improving those symptoms and resolving the problem. For example, in a model built to understand the mechanisms that result in immigration, the model should show how known policies and structure (i.e., causal loops) lead to immigration.

- The *frequency generation test* and *relative phasing test* focus on the relative temporal aspects of relationships between variables, or “on periodicities of fluctuation and phase relationships between variables” (Forrester & Senge, 1980, p. 217). In other words, can the model generate the short- and longer-term fluctuations seen in de real world (frequency generation tests)? And does the relative timing of model output (i.e., the sequential order of values increasing and decreasing) match the relative timing of those variables in the real world (relative phasing tests)? If phase relations obtained from the model are different in the real system, the structure of the model may contain an error. If, for example, the temperature is high, the consumption of gas must be low and vice versa. This must also be reflected in the model behaviour.

- The *multiple-mode test* is especially important in policy analysis, as it investigates whether the simulation model is able to generate more than one mode of observed behaviour. Indeed, a simulation model that is able to produce two or more distinct modes of behaviour which could be observed in a real system (e.g., producing two types of output with the tendency to fluctuate at distinct intervals, for example an economic model that produces 3-year fluctuations and 18-year fluctuations) provides the possibility to study how policies differently affect each of the two modes of behaviour. It is necessary to perform a sensitivity analysis and generate different scenarios with the model to perform this test.

- *Behaviour characteristic tests* are a miscellaneous category for other behaviour reproduction tests that investigate peculiar aspects of model behaviour, e.g., sharp peaks, long troughs, and other unusual events such as a housing bubble or an oil crisis.

Behaviour prediction tests

Behaviour prediction tests are similar to behaviour reproduction tests, but focus on future behaviour, as opposed to observed current or historical behaviour. While SD models do not strive for prediction of future values (no “point prediction”), they should say something about future behaviour. These prediction tests should centre around the conditions leading to a future event or behaviour and on the dynamic nature of the system. In other words, the modeller looks for potential changes – patterns and events – that are likely to occur on the basis of analysis of the model behaviour.

- The *pattern prediction test* qualitatively examines whether the model generates correct patterns of potential future behaviour. A pattern prediction test may include evaluation of periods, phase relationships (consider this the timing of the dynamic behaviour), shape or other characteristics of behaviour simulated by the model.
- The *event prediction test* focuses on particular changes in circumstances, such as a sharp drop in market share or a rapid growth of housing prices.

Behaviour anomaly test

The behaviour anomaly test is extensively used with SD model development. It occurs when the modeller discovers anomalous model behaviour that strongly conflicts with real-world system behaviour. Where does this anomalous behaviour come from? Usually, tracing back to the model elements that cause the anomalous behaviour uncovers obvious flaws in the modelling. However, the *behaviour anomaly test* can also play a role in validation, in addition to model development, since it can be used to defend model assumptions by demonstrating that the model produces peculiar behaviour under different assumptions.

Family member test

SD models have the characteristic that they can be generalised from specific contexts. The *family member test* checks whether this model would also be valid for a slightly different case study. For example, if you consider a population model for the Netherlands, the same structure should be able to generate correct behaviour for other countries, if only the relevant parameter values (e.g., the size of the different population cohorts, and data about fertility and death rates) are adjusted.

Surprise behaviour test

The best and most comprehensive SD models have the potential to generate behaviour that is present in the real system, but which has gone unnoticed by the system's actors. When such unexpected behaviour surprises the modeller, they should aim to understand the underlying causes within the model, and then compare the behaviour and the identified causes to the real-world system. When this procedure leads to identification of previously unrecognized behaviour, the *surprise behaviour test* contributes to confidence in a model study's usefulness.

Extreme conditions test

In extreme conditions tests as part of behavioural validation, you run the model after you changed parameter values to extreme values. These extreme values can be extremely low values (often 0) or extremely high values. You can also test the model behaviour by changing multiple values at the same time to extreme conditions.

The goal is to check whether the behaviour generated by the model under extreme conditions still falls within realistic bounds. For example, if a population sub-model in a model of a country's economy is set to be equal to zero initially, there should be no economy.

Boundary adequacy (behaviour) test

In the previous section we stressed the importance of building confidence in the choice for appropriate model boundaries. The structural boundary adequacy test from Section 8.2.1 should often be extended to include model behaviour analysis. Does the model include the necessary structure to address the issues for which it was designed? Again, the answer to this question is closely related to the model *purpose*. Changes in the model structure that alter the model boundaries can have a very significant impact on the behaviour of the model. Therefore, the modeller conceptualises an additional structure that makes sense (e.g., make an endogenous structure exogenous or vice versa) and that might influence model behaviour. The modeller then compares the model output with and without potential new structure to check its effect. If such additional structures have little effect on the original model behaviour, this strengthens the modeller's confidence that the original model boundary was appropriate.

8.3 Interpretation of results

The interpretation of results is based on a careful selection of graphs of behaviour from certain model variables. A model variable chosen in this selection is often referred to as a “key performance indicator” or KPI. In SD literature, generally graphs are shown which show the behaviour over time for various KPIs. Different KPIs may be combined in one graph, but it is advised to only do so when the units of the KPIs are the same. A special type of graph with multiple runs is the stack graph, in which data of different KPIs is stacked in one graph. It is possible to make these graphs in Vensim or other specialised SD software, but you can also export the run data and make the graphs in other software which allows more freedom in designing the picture (e.g., Figure 8.6).

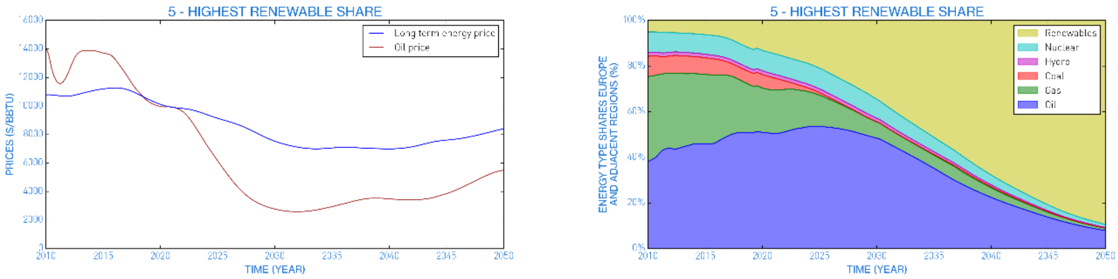


Figure 8.6. Two graphs of a single SD run which were produced in Python: two KPIs showing their behaviour over time in one graph (left) and a stack graph (right).

In the interpretation of results, it is crucial to link the behaviour in the graph to the structure of the model. This can be done by linking behaviour to elements which were, for example, described in the conceptual overview of the model or in the model description. Therefore, it may help to adhere to the following basic questions that one should answer in a figure description:

1. What do we observe in the figure?
2. What behaviour stands out?
3. What structure causes that behaviour?
4. What does that imply?

Example: Evaluation of model behaviour

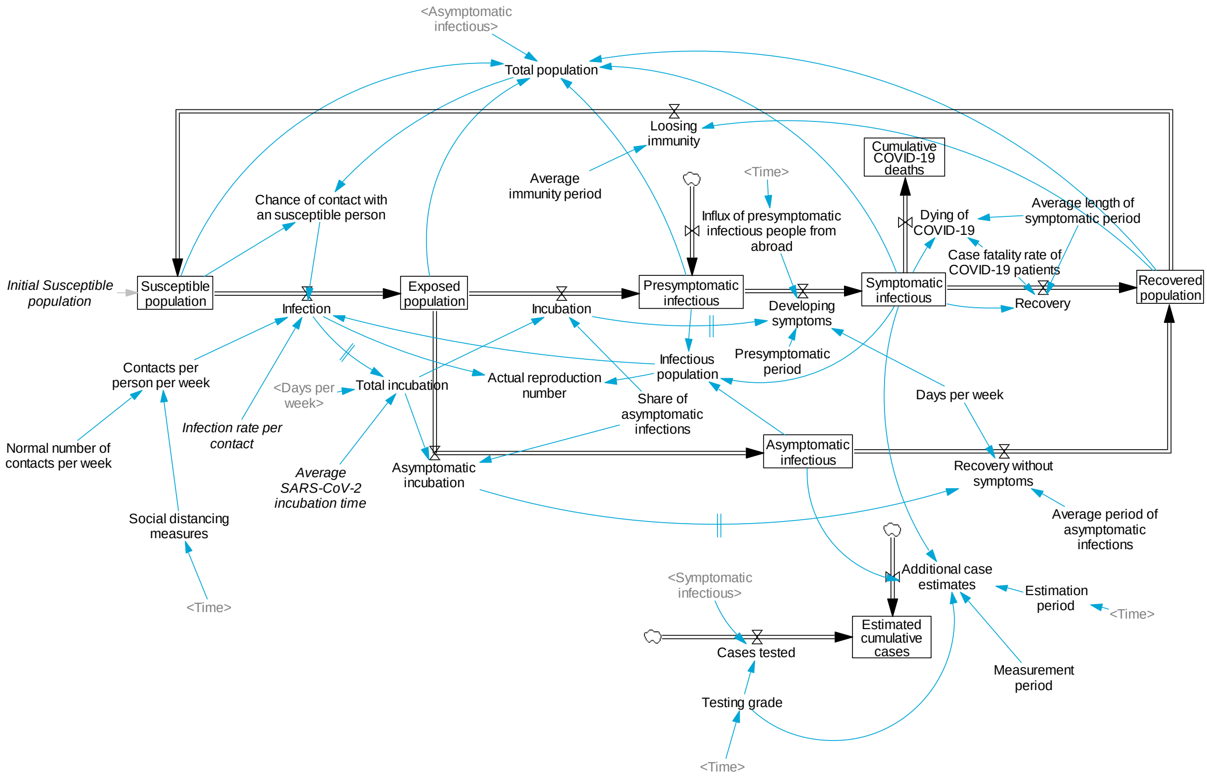


Figure 8.7. Structure of a simple model of the beginning of the COVID-19 pandemic in the Netherlands.

In the evaluation of a model (Figure 8.7, see Exercise 5.8) simulating the early phases of the COVID-19 pandemic in the Netherlands, we observe the behaviour of the variable *infectious population* (Figure 8.8). In this figure, it is striking that we see two “peaks” in the size of the infectious population. This behaviour is mainly caused by three distinctly different structures in the model. First, the *social distancing measures* control the *contacts per person per week*. If people have enough contacts per week, the reinforcing feedback between infectious people and new infections will be dominant. If the number of contacts is relatively low, the balancing feedback between the *susceptible population* and the *total population* makes that the *chance of contact with a susceptible person* is low, and the number of infections decreases. This means that, as expected, social distancing measures play a major role in controlling an epidemic.

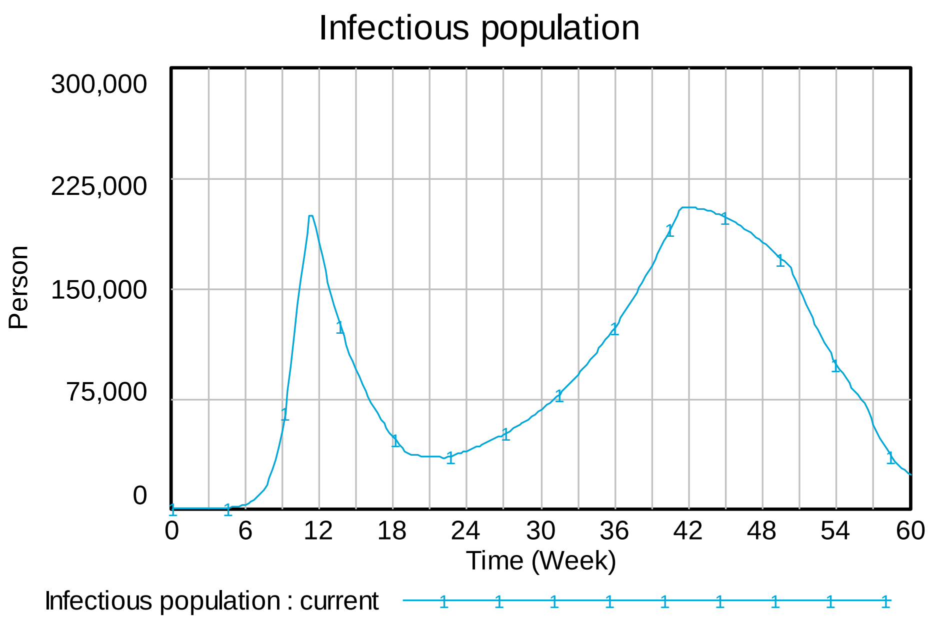


Figure 8.8. Behaviour of the infectious population.

8.4 Scenario development

SD models are frequently used for scenario development. Scenarios can be developed by selecting different values of single (univariate) or multiple (multivariate) input parameters at the same time. The goal of scenario development is to get a broader bandwidth of behavioural modes. These can be used for two reasons. First, they give a better idea of how the system behaviour may plausibly develop. Second, they can be used to test policies for robustness: the policy's ability to keep functioning in the desired way, regardless of how the future unfolds.

A scenario analysis can be performed by hand or with the use of sensitivity analysis tools. If the scenario analysis is performed by hand, a good start is to make a table of all uncertain input parameters. This table lists parameter names, the standard values, units, data sources if applicable or the simple fact that the value is an assumption.

If one wants to get a better idea of how system behaviour may develop, the goal would be to create a diverse set of scenarios. This can be achieved in the following way:

1. Determine which input parameters or input scenarios are uncertain;
2. Determine to which of the uncertain input parameters the KPIs are sensitive;
3. Select two or three uncertain and sensitive parameters and make a scenario logic;
4. Decide to do two or three runs per axis in the scenario logic;

5. Run the model for each different scenario and save the runs with different run names apart from the base case.

If one wants do test policies for robustness, the goal would be to test the policies for each different scenario. Robustness is a policy characteristic which means that the policy functions in the desired way in all plausible futures (Lempert, Groves, Popper, & Bankes, 2006). There are different possible measures for robustness (e.g., see Kwakkel, Eker, & Pruyt, 2016), for example to make sure that the KPIs stay within desirable bounds, or that the results change in a desirable direction in all scenarios.

Example: Scenario development

Scenario development starts with creating an overview of the parameters or driving forces which are at least somewhat uncertain, and might influence model behaviour. Table 8.1 shows this for the COVID-19 pandemic.

Table 8.1. Overview of uncertain parameter values in the COVID-19 model.

Parameter	Unit	Value
Average immunity period	Week	52
Average length of symptomatic period	Week	1
Average period of asymptomatic infections	Day	6
Average SARS-CoV-2 incubation time	Day	3
Case fatality rate of COVID-19 patients	Dimensionless	0.01
Infection rate per contact	Dimensionless	0.30
Normal number of contacts per week	1/week	7.5
Presymptomatic period	Day	3
Share of asymptomatic infections	Dimensionless	0.5

The next step is to perform a univariate sensitivity analysis in order to find out for which of these parameters the model is sensitive. We can do this by changing their values one by one with, for example, plus and minus 10%, and observe the impact on the *infectious population* (Figure 8.9). This implies performing 19 runs in this case, as there are nine uncertain parameters plus the base case.

You can imagine that this job can quickly become very tedious if the model is relatively large. This is also why software like EMA Workbench (Kwakkel, 2017) has been developed, which contains algorithms to deduce in a multivariate way which parameters have a significant impact on the model behaviour. Specifically, what we call “scenario discovery” is aimed at this purpose (Bryant & Lempert, 2010; Kwakkel, Auping, & Pruyt, 2013).

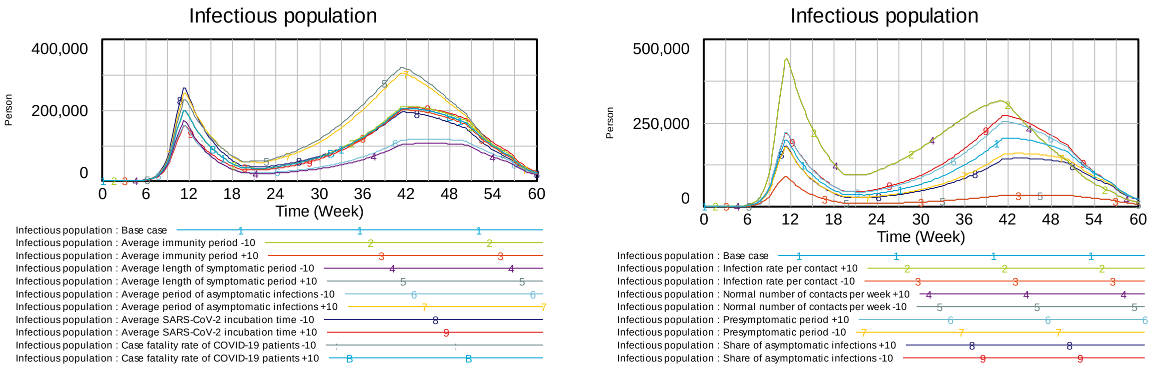


Figure 8.9. The sensitivity of *infectious population* to the different inputs.

Figure 8.9 shows which inputs have most impact on the system, as measured by the variable *infectious population*. If you combine this with existing knowledge of the system, you can determine which inputs are the most uncertain. You can classify these variables in a table (see Table 8.2). In this case, the variables *infection rate per contact* and *normal number of contacts* per week are both uncertain and have a high impact in the univariate sensitivity analysis.

Table 8.2. Classification of driving forces.

		Uncertainty	
		High	Low
Impact	High	*infection rate per contact; normal number of contacts per week*	*average length of symptomatic period; average period of asymptomatic infections*
	Low	*average immunity period; share of asymptomatic infections*	*average SARS-CoV-2 incubation time; case fatality rate of COVID-19 patients; presymptomatic period*

Once we have identified which variables are the most uncertain and have the most impact, we can develop a scenario logic. We will use the *infection rate per contact* and the *normal number of contacts per week* for that logic (see Figure 8.10). We use the logic to explain the experimental setup we can use for performing runs with the model.

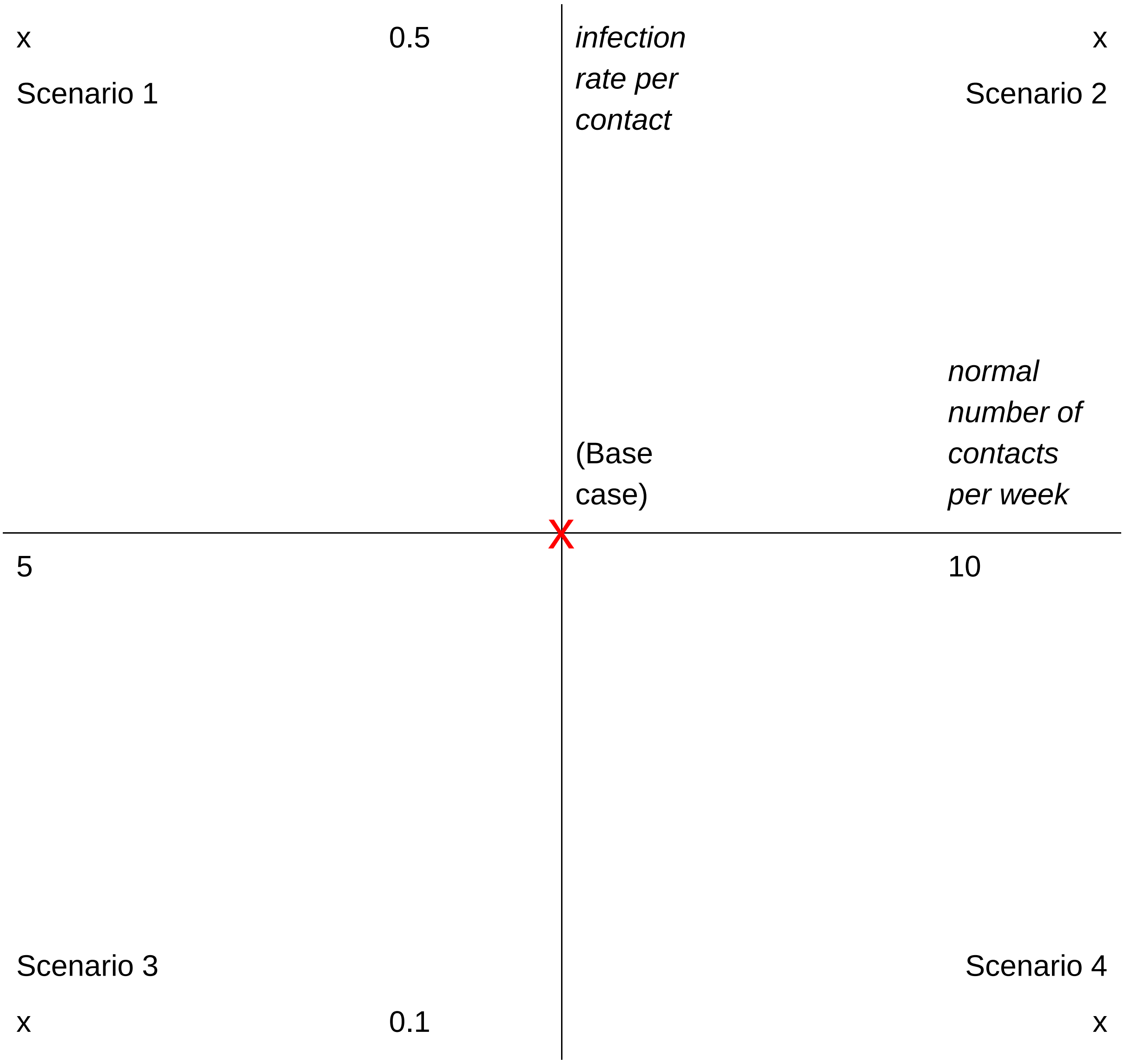


Figure 8.10. Scenario logic for the COVID-19 model.

In this experimental setup, you explain how many runs you perform, the time horizon, the time step and the integration method, and what the parameter values are that feed into these runs. In this case, the experimental setup could contain the following.

> We run the model five times over a time horizon of 60 weeks. We use Vensim Pro version 9.3.3. x64 on a Windows computer. The time step is 0.0078125 weeks and the integration type is Euler due to the discrete nature of the social distancing

measures included in the model. The base case and the scenarios use the following input values (Table 8.3).

Table 8.3. Experimental setup

Scenario	Infection rate per contact	Normal number of contacts per week
Base case	0.3	7.5
Scenario 1	0.5	5
Scenario 2	0.5	10
Scenario 3	0.1	5
Scenario 4	0.1	10

You see that the bandwidth of the parameters for the scenario analysis is broader than the bandwidth used in the sensitivity analysis. This is intentional. The goal of the sensitivity analysis is to show the model's sensitivity to fluctuating different parameters, each with the same relative difference. On the other hand, the goal of the scenario analysis is to show what different yet plausible futures might look like. Therefore, it is necessary to use broader bandwidths.

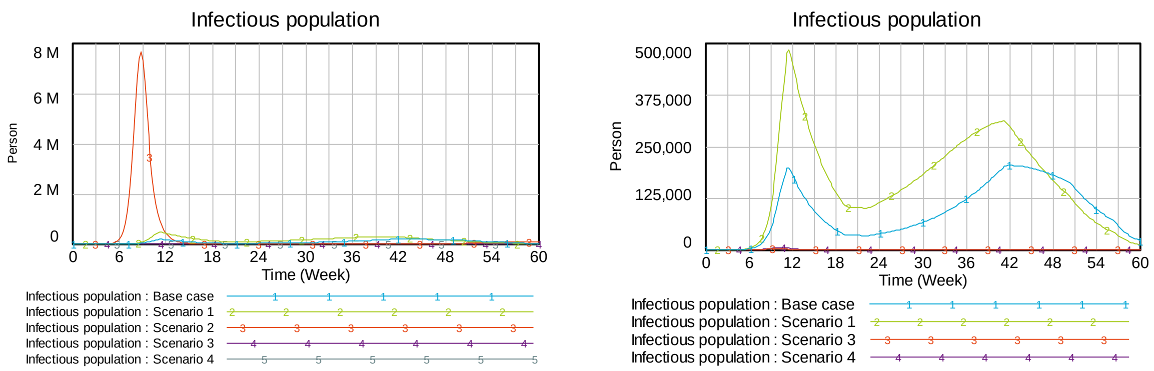


Figure 8.11. Scenario runs with (left) and without (right) scenario 2. Note that the scale of the y axis is different in both graphs.

Figure 8.11 shows the scenario runs. We observe that situations with a high number of contacts and high infectivity lead to a highly undesirable, extremely high first peak of infections. In that situation, the existing social distancing measures are insufficient. Therefore, policies should aim at reducing the case load in those situations (i.e., scenario 2), but they could be less stringent in case of situations with a far lower case load (i.e., scenarios 3 and 4).

In this case, we used a “full factorial” scenario design with two points per scenario axis. As a consequence, you have to perform $2^{\#\text{scenario axes}}$, plus 1 for the base case. You could also choose for a design with more points, for example three, per scenario axis. In that case, the base case becomes one of the scenarios. This eventually leads to a situation in which an ensemble of runs is generated, in which there is no such thing as a base case, but rather a number of plausible futures. The likelihood of those scenarios is not communicated, just that they are all plausible. The base case is then abolished and replaced by a “base ensemble”: the total set of runs before policies are tested.

9. Model use and policy testing

In this chapter, we will discuss how to test policies in SD models. First, we will look at how to design new policies for improving the behaviour of a system or to avoid undesirable futures. These policies may be either static or adaptive. Next, we will discuss how to measure the robustness of policies and how to consider their efficacy and efficiency. Finally, we will briefly consider the termination of policies.

9.1 Policy design

The first step of policy design with SD models is to identify which runs show undesirable behaviour. In different types of models, undesirable futures may be related to different modes of behaviour. For example, values may become untenable. If you consider the case of societal ageing, situations where government spending on retirement funds and healthcare becomes untenable are undesirable. In the case of the COVID-19 pandemic, situations with high mortality or many sick people simultaneously, can cause societal disruption and are undesirable. Policy design will have to alleviate such situations.

The second step is to identify policy levers. These are variables in the model, often constants, that can be influenced by the problem owner. An example of a policy lever for the Dutch government in case of societal ageing is the formal retirement age. Examples of policy levers during the COVID pandemic were the average number of contacts per week, isolation of known positive COVID cases and the introduction of vaccines.

9.1.1 Static policies

Static policies generally rely on changing the value of a parameter that is linked to one of the problem owner's policy levers. Static policies only change as a result of a conscious, deliberate decision. A good example of a static policy – for a very long time – was the retirement age of 65 years in most north-western European countries. Ad hoc policy changes without a predefined rule set to change them can be seen as a series of static policies. Static policies can also be referred to as *open loop policies*, as they do not structurally change the model and do not introduce new feedbacks.

Example: Static policy

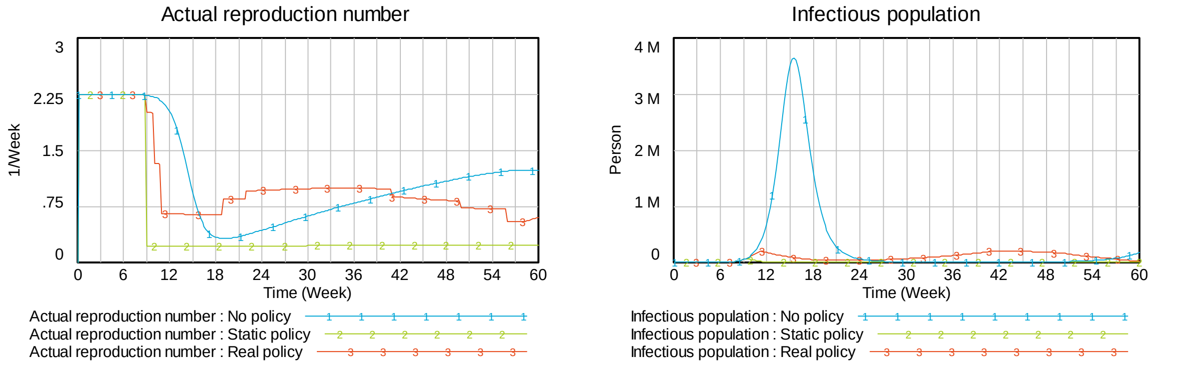


Figure 9.1. Actual reproductive number (left) and infectious population (right) with no policy, a static policy and the real policy derived from the actual situation in the Netherlands in 2020.

In this example, we introduce a very stringent static policy for the case of the COVID-19 pandemic. To do this, we first disconnect the original policy. Figure 9.1 shows, firstly, the behaviour of the model in an uncontrolled epidemic in a completely susceptible population. First there is exponential growth, followed by a situation in which the growth is reduced when a large enough share of the original susceptible population is either exposed, infectious or recovered, and finally a situation in which a new, smaller peak may grow as recovered people lose their immunity.

The static policy with strict social distancing rules is strong enough to avoid the actual reproduction number ever getting over 1. As a consequence, it quickly stops infections from occurring. Policies like this are dangerous, however, as at some moment discontent among the population will grow and demand a relaxation of the policies. This is exactly what happened in China after the immediate suspension of the original, very strict policies. The population was insufficiently immune (i.e., a large share of the population was still susceptible), causing an enormous surge in cases.

9.1.2 Adaptive policies

Adaptive policies rely on changing the system structure by introducing feedback between an existing system variable and a policy lever. A rule linking the performance indicator with the policy lever should be made explicit and, ideally, be institutionally embedded. A good example of an adaptive policy is the change in the formal retirement age in the Netherlands, which was captured (with a formula) in new legislation. As a consequence, the retirement age will gradually rise from 65 to 70 years in 2069. Adaptive policies can be seen as *closed loop policies*, as they add a balancing loop to the system. For the societal ageing case, this is visible in Figure 9.2.

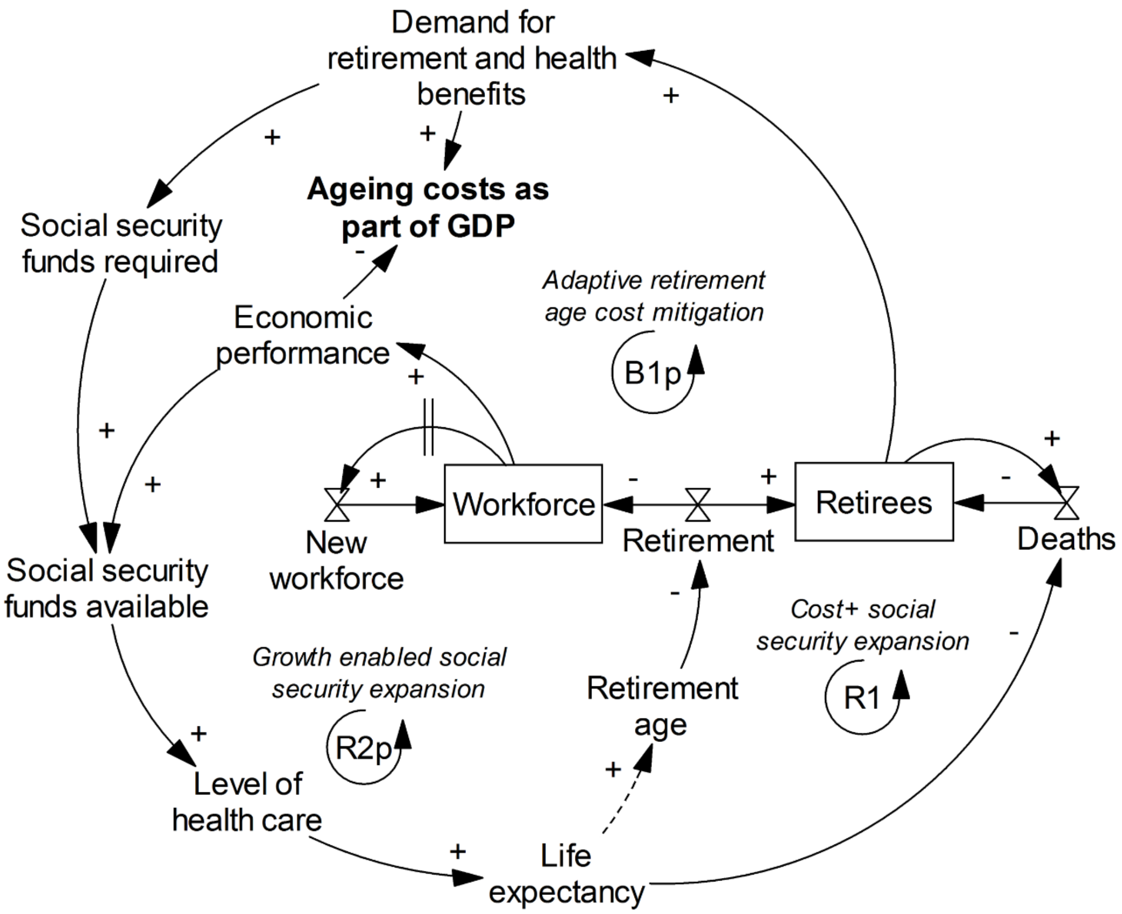


Figure 9.2. Hybrid CLD/SFD of the societal ageing model. Observe how the adaptive retirement age introduces two new loops (R2p and B1p).

Example: Adaptive policies

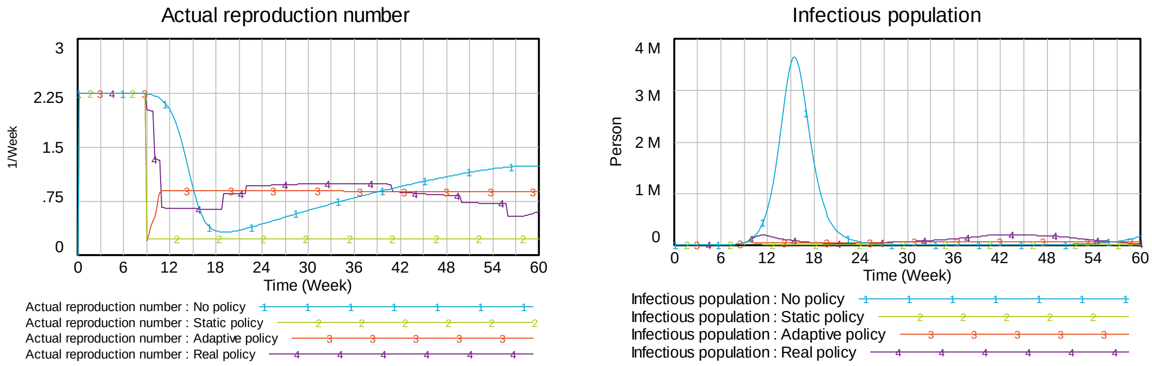


Figure 9.3. Actual reproductive number (left) and infectious population (right) with no, a static, an adaptive policy, and the real policy derived from the actual situation in the Netherlands in 2020.

For the adaptive policy in the case of COVID-19, we introduce a rule to link the *actual reproduction number* to the *number of contacts* the population is allowed to experience. As the severity of the epidemic increases, the reproductive number will become higher than 1 and keep increasing. The higher the reproductive number, the fewer contacts people are allowed, and the lower the reproductive number, the more contacts are allowed.

Figure 9.3 compares the difference between the adaptive policy and a situation with no policy, a static policy and the ad hoc policy derived from reality. It is clear that it is possible to create a situation that allows approximately the same number of contacts as in reality, but does not have to become more stringent after the initial policy change. The adaptive policy will generally lead to less discontent, as it is clearer for the population what to expect. At the same time, the situation does lead to more infections than very stringent static policies.

9.2 Policy robustness

Policies need to function in the desired way under all plausible futures (i.e., scenarios). If policies meet this criterion, we consider them to be “robust”. There are multiple different definitions of robustness criteria, which are called robustness metrics (McPhail et al., 2018). It is, however, possible to distinguish all these different metrics in two main categories. In the first category, the “regret”-based metrics, policies need to have the desired effect in all different scenarios. In the second category, the “satisficing”-based metrics, the policies are considered desirable if they make outcomes fit certain bandwidths.

An example of regret-based robustness is the assessment of policies during a pandemic like COVID-19. Policies aimed at reducing mortality through the disease should do so in all different futures, not just in the worst-case situations. An example of satisficing-based robustness is the effectivity of policies aimed at reducing costs of societal ageing. These policies should reduce the costs in such a way that they are in a sustainable bandwidth for the government. However, it does not matter if in the best-case scenarios the costs increase, as long as it remains within the allowable limits.

Especially for SD models it is relevant to test for policy robustness. Even with minimal changes in inputs, SD models can generate completely different behavioural modes. This is called non-linearity. In all non-linear models, it is unknown whether a model will respond similarly to model inputs – which policies are by definition – if the model is in a different state or at a different moment in the simulated time horizon. Therefore, it is always relevant to check whether policies function in the desired way in a broad range of scenarios.

References

Auping, W. L. (2018). *Modelling Uncertainty: Developing and Using Simulation Models for Exploring the Consequences of Deep Uncertainty in Complex Problems.* (PhD). Delft University of Technology, Delft. Retrieved from https://doi.org/10.4233/uuid:0e0da51a-e2c9-4aa0-80cc-d930b685fc53

Auping, W. L., & d'Hont, F. M. (2023). *A beginners' guide to debugging SD models*. Paper presented at the International System Dynamics Conference, Chicago.

Auping, W. L., d'Hont, F. M., Kubli, M. D., Steinmann, P., Slinger, J., Heijde, F. v. d., . . . Thissen, W. A. H. (2024). The Delft Method for System Dynamics - Modelling Exercises. *SURFsharekit publication.* Retrieved from https://surfsharekit.nl/public/4fab110e-5e08-4265-a3f2-724dffe9022b

Auping, W. L., Pruyt, E., De Jong, S., & Kwakkel, J. H. (2016). The geopolitical impact of the shale revolution: Exploring consequences on energy prices and rentier states. *Energy Policy, 98*(2016), 390-399. doi:10.1016/j.enpol.2016.08.032

Auping, W. L., Pruyt, E., & Kwakkel, J. H. (2017). Simulating endogenous dynamics of intervention-capacity deployment: Ebola outbreak in Liberia. *International Journal of Systems Science: Operations & Logistics, 4*(1), 53-67. doi:10.1080/23302674.2015.1128576

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of Operations Research, 53*(1), 121-173. doi:10.1007/BF02136828

Balci, O. (2013). Verification, Validation, and Testing of Models. In S. I. Gass & M. C. Fu (Eds.), *Encyclopedia of Operations Research and Management Science* (pp. 1618-1627). Boston, MA: Springer US.

Barlas, Y. (1996). Formal aspects of model validity and validation in system dynamics. *System Dynamics Review, 12*(3), 183-210.

Borelli, R. L., & Coleman, C. S. (2004). *Differential Equations. A Modeling Perspective*. New York: John Wiley & Sons, Inc.

Bossel, H. (2007a). *System Zoo 1 Simulation Models: Elementary Systems, Physics, Engineering*. Norderstedt, Germany: Books on Demand GmbH.

Bossel, H. (2007b). *System Zoo 2 Simulation Models: Climate, Ecosystems, Resources*. Norderstedt. Germany: Books on Demand GmbH.

Bossel, H. (2007c). *System Zoo 3 Simulation Models: Economy, Society, Development*. Norderstedt, Germany: Books on Demand GmbH.

Boyce, W. E., & DiPrima, R. C. (2005). *Elementary Differential Equations and Boundary Value Problems* (8th ed.). New York: John Wiley & Sons.

Bryant, B. P., & Lempert, R. J. (2010). Thinking inside the box: A participatory, computer-assisted approach to scenario discovery. *Technological Forecasting and Social Change, 77*(1), 34-49. doi:10.1016/j.techfore.2009.08.002

Carver, R. (1996). Theory for Practice: A Framework for Thinking about Experiential Education. *Journal of Experiential Education, 19*(1), 8-13. doi:10.1177/105382599601900102

Coyle, R. G. (1996). *System Dynamics Modelling. A practical approach*. Dordrecht, Netherlands: Springer-Science+Business Media, B.Y.

De Jong, S., Auping, W. L., & Govers, J. (2014). *The Geopolitics of Shale Gas*. Retrieved from The Hague: http://static.hcss.nl/files/uploads/2180.pdf

Duggan, J. (2016). *System Dynamics Modeling with R*. Switzerland: Springer International Publishing.

Euler, L. (1768). *Institutionum calculi integralis*. Petropoli: Impensis Academiae Imperialis Scientiarum.

Fleming, N. D. (1995). *I'm Different; Not Dumb. Modes of Presentation (VARK) in the Tertiary Classroom.* Paper presented at the Research and Development in Higher Education.

Ford, A. (2009). *Modeling the Environment* (2nd ed.): Island Press.

Forrester, J. W. (1961). *Industrial Dynamics*. Cambridge, MA: MIT Press.

Forrester, J. W. (1969). *Urban Dynamics*: Pegasus Communications.

Forrester, J. W. (1994). System dynamics, systems thinking, and soft OR. *System Dynamics Review, 10*(2-3), 246-256. doi:10.1002/sdr.4260100211

Forrester, J. W., & Senge, P. M. (1980). Tests for Building Confidence in System Dynamics Models. In A. A. Lagasto, J. W. Forrester, & J. M. Lyneis (Eds.), *TIMS Studies in the Management Sciences* (Vol. 14, pp. 209-228). Amsterdam, Netherlands: North-Holland Publishing Company.

Gardner, H. E. (1983). *Frames of mind: The theory of multiple intelligences*. New York, NY: Basic Books.

Hanau, A. (1928). *Die Prognose der Schweinepreise*. Berlin: Verlag von Reimar Hobbing.

Hodges, J. S., & Dewar, J. A. (1992). *Is It You or Your Model Talking? A Framework for Model Validation*. Retrieved from Santa Monica, CA: http://www.rand.org/content/dam/rand/pubs/reports/2006/R4114.pdf

Keating, E. H. (1998). *Everything You Ever Wanted to Know about How to Develop A System Dynamics Model, But Were Afraid to Ask*. Paper presented at the 16th International Conference of the System Dynamics Society, Quebec, Canada. http://www.systemdynamics.org/conferences/1998/PROCEED/00024.PDF

Krugman, P. (2012). *End This Depression Now!* New York, London: W.W. Norton & Company.

Kutta, M. W. (1901). Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Z. Math. Phys., 46*, 435-453.

Kwakkel, J. H. (2017). The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. *Environmental Modelling and Software, 96*, 239-250. doi:10.1016/j.envsoft.2017.06.054

Kwakkel, J. H., Auping, W. L., & Pruyt, E. (2013). Dynamic scenario discovery under deep uncertainty: The future of copper. *Technological Forecasting & Social Change, 80*(4), 789-800. doi:10.1016/j.techfore.2012.09.012

Kwakkel, J. H., Eker, S., & Pruyt, E. (2016). How robust is a robust policy? Comparing alternative robustness metrics for robust decision-making. In *International Series in Operations Research and Management Science* (Vol. 241, pp. 221-237): Springer.

Lane, D. C. (1993). The road not taken: Observing a process of issue selection and model conceptualization. *System Dynamics Review, 9*(3), 239-264. doi:10.1002/sdr.4260090303

Lane, D. C. (1995). *The Folding Star: A comperative reframing and extension of validity concepts in system dynamics*. Paper presented at the Procedings of the 1995 International System Dynamics Conference, Tokyo, Japan.

Lane, D. C. (2000). Diagramming conventions in system dynamics. *Journal of the Operational Research Society, 51*(2), 241-245. doi:10.1057/palgrave.jors.2600864

Lempert, R. J., Groves, D. G., Popper, S. W., & Bankes, S. C. (2006). A General, Analytic Method for Generating Robust Strategies and Narrative Scenarios. *Management Science, 52*(4), 514-528. doi:10.1287/mnsc.1050.0472

Martinez-Moyano, I. J. (2012). Documentation for model transparency. *System Dynamics Review, 28*(2), 199-208. doi:10.1002/sdr.1471

Mayer, J., Khairy, K., & Howard, J. (2010). Drawing an elephant with four complex parameters. *American Journal of Physics, 78*(6), 648-649. doi:10.1119/1.3254017

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics, 21*(2), 239-245. doi:10.2307/1268522

McPhail, C., Maier, H. R., Kwakkel, J. H., Giuliani, M., Castelletti, A., & Westra, S. (2018). Robustness Metrics: How Are They Calculated, When Should They Be Used and Why Do They Give Different Results? *Earth's Future, 6*(2), 169-191. doi:10.1002/2017EF000649

Meadows, D. H. (1976, 1976). *The Unavoidable A Priori.* Paper presented at the International Conference on System Dynamics.

Meadows, D. H. (2008). *Thinking in Systems. A Primer*. London, UK: Earthscan.

Meadows, D. H., Meadows, D. L., Randers, J., & Behrens, W. W., III. (1972). *The Limits to Growth: A Report for the Club of Rome's Project on the Predicament of Mankind*. New York: Universe Books.

Morecroft, J. D. W. (1982). A Critical Review of Diagramming Tools for Conceptualizing Feedback System Models. *Dynamica, 8*(1), 20-29.

Morecroft, J. D. W. (2015). *Strategic Modelling and Business Dynamics: A feedback systems approach* (2nd ed.): John Wiley & Sons.

Nordhaus, W. D. (1973). World Dynamics: Measurement Without Data. *The Economic Journal, 83*(332), 1156-1183. doi:10.2307/2230846

Oreskes, N., Shrader-Frechette, K., & Belitz, K. (1994). Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences. *Science Magazine, 263*(5147), 641-646. doi:10.1126/science.263.5147.641

Pruyt, E. (2013). *Small System Dynamics Models for Big Issues: Triple Jump towards Real-World Complexity*.

Randers, J. (Ed.) (1980). *Elements of the System Dynamics Method*. Cambridge, MA: Productivity Press.

Richardson, G. P., & Pugh, A. L., III. (1981). *Introduction to System Dynamics Modeling with Dynamo*. Cambridge, MA: MIT Press.

Roberts, E. B. (1978). *Managerial Applications of System Dynamics*. In E. B. Roberts (Ed.).

Runge, C. D. T. (1895). Über die numerische Auflösung von Differentialgleichungen. *Math. Ann., 46*, 167-178.

Sciarone, A. G., & Montens, F. (1985). Nederlands Voor Buitenlanders. *Toegepaste Taalwetenschap in Artikelen, 22*, 156-165. doi:10.1075/ttwia.22.13sci

Senge, P. M. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY: Doubleday/Currency.

Sterman, J. D. (1994). Learning in and about complex systems. *System Dynamics Review, 10*(2-3), 291-330. doi:10.1002/sdr.4260100214

Sterman, J. D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York: McGraw.

Strang, G., & Herman, E. J. (2016). *Calculus Volume 2*. Houston, TX.

Van Daalen, C. E., Thissen, W. A. H., & Phaff, H. W. G. (2006). *spm2310/epa1321 Continuous Systems Modelling/System Dynamics*. Delft: Technology, Policy and Management.

Ventana Systems. (2010). *Vensim Reference Manual*. Harvard, MA: Ventana Systems.

Warren, K. (2002). *Competitive Strategy Dynamics*: John Wiley & Sons.

Wolstenholme, E. F. (1989a). Applying System Dynamics. *Transactions of the Institute of Measurement and Control, 11*(4), 170-179. doi:10.1177/014233128901100401

Wolstenholme, E. F. (1989b). *System Enquiry: a System Dynamics Approach*. Chichester: John Wiley & Sons.

The Delft Method for System Dynamics

Willem Auping, Floortje d'Hont, Merla Kubli, Jill Slinger, Patrick Steinmann, Floris van der Heijde, Els van Daalen, Erik Pruyt, Wil Thissen

The Delft Method for System Dynamics (SD) is a proven method of learning basic SD. The method focusses on learning by doing: first you try to make an exercise, and if you do not understand something or do not know something, then you can look it up in the theory part of the book. The book contains theory exercises on topics like causal loop diagramming, delays, and how to motivate why SD is appropriate. The book also contains modelling exercises that show students how to build low to medium complexity models, and how to use these quantitative models for scenario or policy analysis. The theory chapters cover all phases of the modelling cycle: problem articulation, conceptualisation, formulation, evaluation (including validation and scenario analysis), and policy analysis. This book is intended for students and educators in large or small System Dynamics courses, and for motivated students that want to learn SD in their own pace.

The System Dynamics education team of TU Delft developed and published "The Delft Method for System Dynamics". The material builds on decades of System Dynamic teaching experience at Delft University of Technology in the Netherlands, where Bachelor and Master students are trained in various methods for modelling and simulating complex grand challenges (Pruyt, 2013; Van Daalen et al., 2006). All co-authors have contributed to the development of System Dynamics education at TU Delft.

Willem Auping is a methodologist specialised in the impact deep uncertainty has on the way we develop and use simulation models.

Floortje d'Hont is a policy analyst with a particular interest for the role of systems thinking in education, transdisciplinary research and participation.

Merla Kubli is a systems researcher specialised in the business dynamics of climate solutions. She combines choice experiments with System Dynamics simulation to investigate consumer-provider interactions for emerging business models.

Jill Slinger is a transdisciplinary water and coastal governance specialist, who held many key appointments at the interface between science and policy. Her expertise builds on mathematical and simulation modelling, field-based research on aquatic systems, and engagement with communities and policy makers.

Patrick Steinmann, a decision support scientist, specialises in using simulation-based methods to enhance foresight and decision-making under deep uncertainty, particularly in the safety and security domain.

Floris van der Heijde is a Master student Engineering & Policy Analysis at TU Delft, interested in simulation modelling and modelling supported decision-making. He was also actively involved in teaching activities throughout his master's.

Els van Daalen is a retired associate professor of policy analysis at TU Delft, interested in problem structuring, systems modelling, and serious gaming.

Erik Pruyt is a systems modeller and director of the Center for Policy Exploration, Analysis and Simulation, an organization that specialises in modelling complex issues and simulating them under uncertainty.

Wil Thissen is an emeritus professor interested in applying systems methodology to public decision problems.

TU Delft

ISBN 978-94-6366-894-1
DOI: https://doi.org/10.59490/tb.97
textbooks.open.tudelft.nl

Cover image: Willem Auping