Chapter 3

System dynamics and world models

System dynamics has its roots in systems of difference and differential equations (Forrester 1980: Section 3.3). A target system, with its properties and dynamics, is described using a system of equations which derive the future state of the target system from its actual state. System dynamics is restricted to the macro level in that it models a part of reality (the 'target system') as an undifferentiated whole, whose properties are then described with a multitude of attributes in the form of 'level' and 'rate' variables representing the state of the whole target system and its changes, respectively.

The typical *difference* equation has the form

$$x_{t+1} = f(x_t; \vartheta) \tag{3.1}$$

where x_{t+1} is the state of the target system at time $t + 1$, which depends on its state at time t and on a parameter ϑ. Both x and ϑ may be vectors, that is, consist of several elements. f is usually a continuous function. Only in rare cases can the difference equation be solved explicitly to yield an expression for x_t as a function of t and x_0.

The typical *differential* equation has the form

$$\dot{x}(t) = \frac{dx}{dt} = g(x(t); \vartheta) \tag{3.2}$$

where $\dot{x}(t)$ is the state change of the target system within an infinitesimally short period of time dt. The amount of change depends on the state $x(t)$ at

time t and on a parameter ϑ. Again, both x and ϑ may be vectors, and g is usually a continuous function. In simple cases, the differential equation can be solved explicitly, yielding an expression for $x(t)$ as a function of t.

Conceptually, there is a close relationship between difference and differential equations. In the case of difference equations, equidistant points of time are numbered or labelled by t, and nothing is said about the time scale. Hence, we could introduce a new time scale τ_t in which the distance of consecutively labelled or numbered points of time is $\Delta\tau$. If the right-hand side of a difference equation can be written in the following form:

$$x_{t+1} = f(x_t; \vartheta) = x_t + g(x_t; \vartheta) \tag{3.3}$$

meaning that the state at time $t + 1$ is equal to the state at time t, plus the change of state, or with the explicit distance Δt between points of time,

$$x_{\tau_t+\Delta\tau} = f(x_{\tau_t}; \vartheta) = x_{\tau_t} + \Delta\tau \cdot g(x_{\tau_t}; \vartheta) \tag{3.4}$$

(which is always possible), then the following transformation can be performed:

$$x_{\tau_t+\Delta\tau} - x_{\tau_t} = \Delta\tau \cdot g(x_{\tau_t}; \vartheta) \tag{3.5}$$

$$\frac{x(\tau + \Delta\tau) - x(\tau)}{\Delta\tau} = g\left(x(\tau); \vartheta\right) \tag{3.6}$$

Taking limits – that is, as $\Delta\tau$ is reduced to an infinitesimally short period of time ($\Delta\tau \to 0$) – we arrive at

$$\lim_{\Delta\tau\to 0} \frac{x(\tau + \Delta\tau) - x(\tau)}{\Delta\tau} = \frac{dx}{d\tau} = \dot{x}_\tau = g\left(x(\tau); \vartheta\right) \tag{3.7}$$

which is a differential equation. Note that the solution of a differential equation will be different from the solution of the corresponding difference equation. The simplest procedure for finding numerical solutions to differential equations uses the similarity between the two types of equations and a fixed Δt to approximate the differential equation. And this is exactly what system dynamics does, too. Thus, system dynamics differs from systems of differential equations mostly in two technical aspects: discrete time is used as a coarse approximation for continuous time to achieve numerical solutions; and functions of all kinds, not just continuous functions, can be used.

System dynamics also provides the modeller with a graphical description language, the system dynamics diagrams that describe the interdependencies between the attributes of the target system. The graphical symbols – see Figure 3.1 – are taken from the world of streaming water or steam which flows between containers controlled by valves: heating is a favourite example for explaining the principles of feedback loops, and words referring to bonding relations (Bunge 1979) are derived from words used for the same target systems in many languages (for example, 'influence', according to *Webster's Dictionary* was originally 'an ethereal fluid held to flow from the stars and to affect the actions of humans').

Figure 3.1: System dynamics diagram (redrawn from Forrester 1980: Fig. 2.2a)

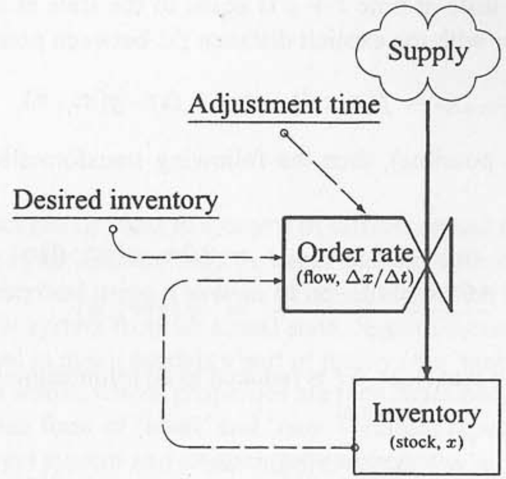


Figure 3.1 shows the supply flow (——) from the inexhaustible cloud (source) into the 'inventory' through the valve 'order rate' which is controlled (– – –) by the actual 'inventory', the 'desired inventory', and the 'adjustment time'. Figures of the same kind may also be used to visualize the control of more complex feedback loops, as in the case of models of the dynamics of the world system. Such complex target systems and their models show, however, that there are limits to the system dynamics diagram technique: a diagram measuring 60 cm by 40 cm with a barely decipherable legend (as on the back flap of Meadows *et al.* 1974) is hardly appropriate to communicate an overview. This is because a whole system dynamics model is represented by one single object with a vast number of attributes.

Software

DYNAMO was the first language especially designed for building system dynamics models. It is a functional simulation language that can handle an arbitrary number of equations for:

- levels – for example, `L inventory.k=inventory.j +dt*orderRate.jk`
- rates – for example, `R orderRate.kl= (desiredInventory-inventory.k)/adjustmentTime`
- constants – for example, `C desiredInventory=6000`
- initializations – for example, `N inventory=1000`

Auxiliaries can also be used as a shorthand for complicated expressions, as in the example below (see p. 39).

Over the years, a number of DYNAMO-like simulation languages and simulation systems have been developed. The best known of them include:

- Professional DYNAMO Plus™;
- STELLA, originally developed for Macintosh, and much like DYNAMO, but with important additional features, including a graphical user interface (`http://www.hps-inc.com/`);
- PowerSim (`http://www.powersim.no` or `http://www.powersim.com`) is also equipped with a graphical user interface and allows for all types of system dynamics modelling.
- VenSim (`http://www.vensim.com`) comes in a so-called 'personal learning edition' that 'gets you started in system dynamics modeling' (quoted from the website) as well as standard and professional editions which allow for more complex models as well as for sensitivity analyses.

There are several other packages running DYNAMO or DYNAMO-like languages.

A DYNAMO program consists of expressions that are bound to names. Names do not refer to memory locations where values of variables are stored, but refer to the expressions to which they are bound. The DYNAMO interpreter will evaluate expressions at the time they are first used and store the result of the evaluation for further use. This is why the order of equations in a DYNAMO program is arbitrary (although it is good programming style to start with level equations and initializations of level variables, then place rate equations just below, have equations for auxiliaries follow and end up with constants).

In the example above, the first expression to be evaluated is `inventory.k`, the value that the level `inventory` will assume at this point in time (which always is marked by the suffix `k`). The related expression first contains `inventory.j`, the value that the level assumed at the former point in time (which is always marked by the suffix `j`) – this value will be known from earlier computations and, if not, will be taken from the initialization. In this case the initialization expression `inventory = 1000` will be evaluated to `1000`, where this branch of the evaluation will terminate. The next term in the expression for `inventory.k` consists of two factors, namely `dt` and `orderRate.jk`. `dt` means the length of one time step. The other factor `orderRate.jk` is a rate to which a rate expression is bound. The suffix `jk` denotes the fact that `orderRate.jk` is the rate of flow between `j` and `k`. Thus, `inventory.k` can be assigned a value.

The next step in the evaluation is the rate `orderRate.kl`, the rate of flow during the *next* time step (between `k` and `l`). Expressions for rates may contain references to the values of levels because these are either known or can easily be evaluated (as they must only depend on former values, `.j` and `.jk`). Expressions for auxiliaries are evaluated in the same manner. Auxiliaries, too, have a former and an actual value.

At the end of all evaluations for one point in time, all values of levels, auxiliaries and rates (with suffixes `.k` and `.kl`, respectively) replace the former values (possibly after these have been written to some output file). This means that at every point in time, only the rate and level values of the immediate past are accessible and values about the earlier past are lost. Special functions (for example, the `delay` function) are necessary to model influences from the remote past.

An example: doves, hawks and law-abiders

A differential equation model ...

For an introductory example we take a model that was described by Martinez Coll, who tried 'to develop a formal model of the Hobbesian state of nature from the perspective of bioeconomics' (Martinez Coll 1986: 494). He defines Hobbes' state of nature as a society whose members are continually competing with each other to obtain a resource. All resources belong to someone, thus conflicts arise between resource owners and those who want an additional resource. Martinez Coll follows Maynard Smith (1982) in that he endows the members of his model society with one of three strategies: the hawk, the dove and the law-abiding strategies.[1]

- The *dove* never tries to get hold of others' possessions, but waits until they are given up, and himself abandons his resource as soon as he is attacked. If two compete for the same resource, one of them gets it (through persistence or luck) with equal probability.
- The *hawk* always tries to get hold of others' resources by means of aggression and gives up only if he receives serious injuries.
- The *law-abider* never tries to get hold of others' possessions, but waits until they are given up, and he defends his possession by counterattack until he either succeeds or is defeated.

In Hobbes' state of nature, the human population consists only of hawks, and in Hobbes' 'Commonwealth' only of law-abiders.

The strategies applied by the individuals may spread all over the population, by inheritance, imitation or education. In any case, in a situation defined by the distribution of strategies, the most profitable strategy is transmitted to other members of the population.

To operationalize what a profitable strategy is, we have to make some assumptions about the 'costs' and 'gains' of strategies. We assume that if an individual following strategy i (a hawk, dove or law-abiding strategy) meets an individual following strategy j, i's gain will be r_{ij} (if r_{ij} is negative, i makes a loss in the encounter). The values r_{ij} are given by the utility of possession minus the costs of the fight. Let the utility of possession be u (`poss` in the DYNAMO model), and the costs of fighting or waiting be c_H and c_D (`coha` and `codo`), respectively, and let $c_D < u < c_H$.

Thus, when an individual applying the strategy of any row of the table below meets an individual applying the strategy of any column, they receive the gains shown in the entries of the table (the first term is the gain of the 'row' individual, the second is the gain of the 'column' individual).

	Dove	Hawk	Law-abider
Dove	$\frac{u}{2} - c_D, \frac{u}{2} - c_D$	$0, u$	$\frac{0+\frac{u}{2}-c_D}{2}, \frac{u+\frac{u}{2}-c_D}{2}$
Hawk	$u, 0$	$\frac{u-c_H}{2}, \frac{u-c_H}{2}$	$\frac{\frac{u-c_H}{2}+u}{2}, \frac{\frac{u-c_H}{2}+0}{2}$
Law-abider	$\frac{u+\frac{u}{2}-c_D}{2}, \frac{0+\frac{u}{2}-c_D}{2}$	$\frac{\frac{u-c_H}{2}+0}{2}, \frac{\frac{u-c_H}{2}+u}{2}$	$\frac{u}{2}, \frac{u}{2}$

Division by 2 is interpreted as follows:

[1] We will return to a very similar model in later chapters of this book (Chapter 6, p. 123; see also Werner and Davis 1997).

- When two individuals applying the same strategy meet, each of them has the same chance of winning or losing. For example, if two hawks meet, one of them will get the resource (u), while the other will receive serious injuries ($-c_H$). Since both have the same chance of winning, the expected outcome will be $\frac{u-c_H}{2}$.
- When a law-abider meets another individual each of them may be the lawful owner of the resource competed for. For example, if a dove meets a law-abider and both compete for the same resource, then we have two equally probable possibilities:
  - If the law-abider is the lawful owner of the resource, it keeps the resource (u), and the dove takes nothing (0).
  - If the dove is the lawful owner of the resource, both have to wait until one of them gives up ($-c_D$) and then one of them gets the resource with equal probability, so the expected outcome of this case is $\frac{u}{2} - c_D$ for both of them.

  Thus the overall outcome is $\frac{u+(\frac{u}{2}-c_D)}{2}$ for the law-abider and $\frac{0+(\frac{u}{2}-c_D)}{2}$ for the dove.

For our numerical example, we will take the following numbers: $c_D = 3$, $u = 10$ and $c_H = 20$, which yields the following payoff matrix:

	Dove	Hawk	Law-abider
Dove	2, 2	0, 10	1, 6
Hawk	10, 0	–5, –5	2.5, –2.5
Law-abider	6, 1	–2.5, 2.5	5, 5

Now we have to observe the average gain $y_i(t)$ of an individual applying strategy i at time t: it is given by the mean of possible gains, weighted by the proportions p_i of the population following each of the strategies, i:

$$y_i(t) = \sum_j r_{ij} p_j(t) \tag{3.8}$$

This average gain of strategy i must be compared with the mean gain of all strategies:

$$y(t) = \sum_i y_i(t) p_i(t) \tag{3.9}$$

The growth of the subpopulation applying strategy i is modelled as proportional to the difference $F_i(t)$ between its average gain and the overall mean gain of all strategies $y(t)$:

$$F_i(t) = y_i(t) - y(t) \tag{3.10}$$

If $F_i(t)$ is positive, then strategy i is more successful than the average and it is inherited, imitated or indoctrinated more often; that is, it spreads faster than the overall mean of the strategies. Thus, the relative growth of the strategies can be written as follows:

$$p_i(t+1) = p_i(t)[1 + F_i(t)] \tag{3.11}$$

This difference equation can be transformed into a differential equation if we assume that within a time span of length Δt the effects on growth are reduced by this factor (compare equations (3.3)–(3.7)):

$$p_i(t+\Delta t) = p_i(t)[1 + \Delta t F_i(t)] \tag{3.12}$$

$$\frac{p_i(t+\Delta t) - p_i(t)}{\Delta t} = p_i(t) F_i(t) \tag{3.13}$$

Taking limits, we have

$$\lim_{\Delta t \to 0} \frac{p_i(t+\Delta t) - p_i(t)}{\Delta t} = \dot{p}_i = p_i F_i \tag{3.14}$$

Inserting equations (3.10), (3.9) and (3.8) into equation (3.14) yields the relative growth of strategy i:

$$\dot{p}_i = p_i F_i \tag{3.15}$$

$$= p_i(y_i - y) \tag{3.16}$$

$$= p_i y_i - p_i \sum_k y_k p_k \tag{3.17}$$

$$= p_i \sum_j r_{ij} p_j(t) - p_i \sum_k \sum_j r_{kj} p_j(t) p_k(t) \tag{3.18}$$

which is a cubic differential equation of the same type as described by Eigen and Schuster (1979: 30–31) (selection under constrained growth with nonlinear growth rates) and used by Troitzsch (1994: 44).

... and its analytical treatment

Differential equation models of this type can be treated in three different ways:

- by linear stability analysis, where interest centres on whether the model can assume a stationary state (or equilibrium, a state in which the system will remain once it has reached this state) and how the system performs in an infinitesimally small region of its phase space around stationary states, that is, whether the equilibria are stable or unstable;

- by global stability analysis, which is concerned with whether stationary states are attractors or repellors, that is, whether the system approaches or escapes stationary states from arbitrary initial states;
- by numerical treatment, in which a large number of trajectories are calculated starting from different initial states.

The first question is whether and where a system has stationary states. This is addressed by determining those states in which the right-hand sides of the system of differential equations become zero. In these states, the derivatives, that is, the time-dependent changes of all state variables, are zero, and consequently, the system will remain one of these states once it has been reached. This means we equate the right-hand side of equation (3.18) to zero:

$$0 = p_i \sum_j r_{ij} p_j(t) - p_i \sum_k \sum_j r_{kj} p_j(t) p_k(t) \tag{3.19}$$

Three first candidates for stationary states are all the states in which the whole population applies the same strategy. For $p_D = p_H = 0$ and, consequently, $p_L = 1$, equation (3.19) is satisfied for $i = D$ (dove) and $i = H$ (hawk); and for $i = L$ (law-abider) it simplifies to

$$0 = 1 \cdot r_{LL} \cdot 1 - 1 \cdot r_{LL} \cdot 1 \cdot 1 \tag{3.20}$$

and the same is true for all permutations of indices.

There is a fourth stationary state, in which doves and hawks coexist and law-abiders are absent. To find this stationary state (and to do some mathematical derivations, which are necessary for the following discussion) it is convenient to express the system of differential equations in terms of the constants c_H, c_D and u, and to keep in mind that there are only two coupled differential equations, because at all times $p_L = 1 - p_D - p_H$. By several intricate transformations and insertions, this leads to the following system of differential equations:

$$\begin{aligned} \dot{p}_D &= -\frac{p_H p_D^2}{2}(2c_D + c_H) + \frac{p_H p_D}{4}(2c_H + 2c_D - u) + \\ &\quad + \frac{p_D^2}{4}(2c_D + u) - \frac{p_D}{4}(2c_D + u) \end{aligned} \tag{3.21}$$

$$\begin{aligned} \dot{p}_H &= -\frac{p_H^2 p_D}{2}(2c_D + c_H) + \frac{p_H^2}{4}(c_H - u) + \\ &\quad + \frac{p_H p_D}{4}(4c_D + c_H + u) - \frac{p_H}{4}(c_H - u) \end{aligned} \tag{3.22}$$

Both right-hand sides of this system reduce to zero for

$$p_D = \frac{c_H - u}{2c_D + c_H} \qquad p_H = \frac{2c_D + u}{2c_D + c_H} \tag{3.23}$$

This means that the system will be in equilibrium if the proportion of doves in the population is $\frac{c_H - u}{2c_D + c_H}$ and the proportion of hawks is $\frac{2c_D + u}{2c_D + c_H}$ (and no law-abiders are present).

To find out what happens in an immediate (infinitesimal) neighbourhood of the stationary states we have to approximate the nonlinear system of differential equations (3.21) and (3.22) by a linear system. We leave this analysis to Appendix B (p. 267) which will also give a first introduction to the analytical treatment of equation-based models. Its result is that the only stable state is the state with only law-abiders surviving. The states with only doves, with only hawks, and the mixed state in equation (3.23) are all unstable, so that even minimal fluctuations that import a small fraction of law-abiders into the population will lead to an ever growing proportion of law-abiders. A population starting with an arbitrary mixture of only hawks and doves into which some law-abiders are inserted will first approach the mixed stationary state of equation (3.23) and then the proportion of law-abiders will grow until the law-abiders have driven out all the hawks.

Intuitively, we may assume that the law-abiders are fitter than both hawks and doves. They avoid the additional costs of fighting which the hawks have to bear when they attack others, and they avoid the unnecessary losses the doves have to bear when they do not defend their possessions against attacks by hawks. In a world with a large majority of hawks, law-abiders are not much better off than hawks, because they will behave much like hawks in most encounters (in that they at least start counterattacks), and in a world with a large majority of doves, law-abiders are not much better off than doves, because they will behave like doves in most encounters (in that they wait for a possession until it is given up). But in a mixed world they enjoy their adaptive strategy: in encounters with hawks they have a better expected outcome than doves because they give up less easily than doves, and in encounters with doves they have the better expected outcome than doves because they take the resources away without waiting.

A DYNAMO model

The equations with which we described our model mathematically, (3.8)–(3.11), can easily be transformed into a DYNAMO model. The

Table 3.1: Correspondence between the system of differential equations and the DYNAMO code

$p_D(t+1)$
$p_H(t+1)$
$p_L(t+1)$
	`dove.k`
	`hawk.k`
	`lawa.k`

$p_D(t)$ ...	`dove.j` ...
$y_D(t) = \sum_j r_{Dj} p_j(t)$ ...	`dove.k*rdd+hawk.k*rdh+lawa.k*rdl` ...
$y_D(t) p_D(t)$ ...	`yieldd.kl=(dove.k*rdd+hawk.k*rdh +lawa.k*rdl)*dove.k` ...
$y(t)$	`yields.kl=yieldd.kl+yieldh.kl +yieldl.kl`
$p_D(t) F_D(t) = p_D(t)(y_D(t) - y(t))$	`yieldd.jk-dove.j*yields.jk`

correspondence between the mathematical formulation and the DYNAMO code is given in Table 3.1 (i is replaced by D, H and L, respectively).

Thus, we arrive at a first formulation of the DYNAMO model:

```
dove.k=dove.j+dt*(yieldd.jk-dove.j*yields.jk)
hawk.k=hawk.j+dt*(yieldh.jk-hawk.j*yields.jk)
lawa.k=lawa.j+dt*(yieldl.jk-lawa.j*yields.jk)
yieldd.kl=(dove.k*rdd+hawk.k*rdh+lawa.k*rdl)*dove.k
yieldh.kl=(dove.k*rhd+hawk.k*rhh+lawa.k*rhl)*hawk.k
yieldl.kl=(dove.k*rld+hawk.k*rlh+lawa.k*rll)*lawa.k
yields.kl=yieldd.kl+yieldh.kl+yieldl.kl
...
```

This DYNAMO program is correct, but it does not reflect the fact that the sum of the level variables (`dove.k+hawk.k+lawa.k`) always remains constant. In a population of constant size, there are no flows to and from outside, but only flows among the subpopulations. Observe, however, that a direct flow between the doves' and the law-abiders' populations need not be explicitly modelled. Only net flows via the rates for the doves' and the law-abiders' populations can and need be modelled, since Martinez Coll's explication of his model does not give any clue to the individual flows between the subpopulations. His description is only about growing and shrinking subpopulations, not about individuals changing their strategies – hence we cannot determine how many individuals (or which proportion) 'flow' from, for example, `dove` to `lawa`.

To visualize this fact, one would need a system dynamics diagram without sources and sinks,[2] like that in Figure 3.2 – which, however, is not a systems dynamics diagram in the sense of Forrester, but a diagram that is generated in the first step of STELLA modelling.

A STELLA model

With the help of the STELLA software one does not start with equations, but with graphic symbols that are arranged on screen to yield a diagram that is much like the diagrams invented by Forrester. The STELLA software then converts the diagram into program code, which is similar to, but not identical with DYNAMO code. The main difference between the two formalisms is that STELLA uses a more mathematical notation instead of the cryptic JKL denotation of the time points – as is shown in the program code below.

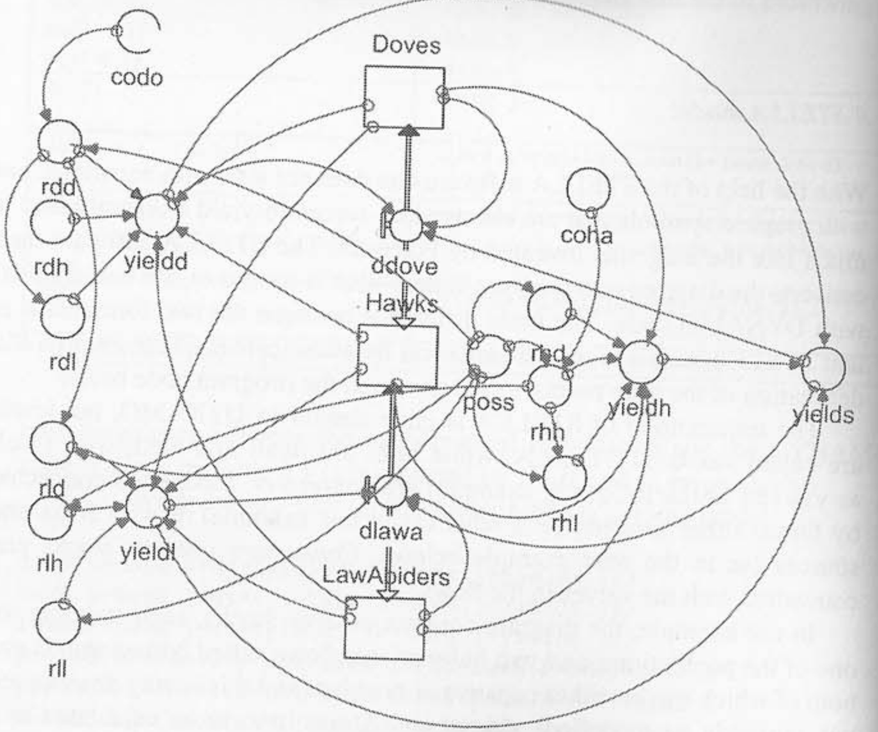
The terminology of STELLA is quite similar to DYNAMO, but levels are called *stocks* in STELLA, while rates are *flows* and auxiliaries (such as `yieldd` in the following example) are *converters*. Stocks are connected by flows, either between each other (as in this example) or with sinks and sources (as in the next example below). *Connectors* connect stocks and converters with the valves in the flows.

In our example, the diagram consists of three stocks, each standing for one of the populations, and two bidirectional flows called `ddove` and `dlawa` both of which can be either negative or positive (and this is why `dhawk` need not explicitly be modelled). `ddove` and `dlawa` have to be calculated in a way that reproduces Martinez Coll's original ideas (see the program code generated by STELLA). Of course, STELLA cannot formulate the right hand sides for `ddove`, for example, instead the STELLA user is given a chance to write down this right hand side. The panel popping up when the `ddove` line in STELLA's code window is double-clicked (see Figure 3.3) lists the required inputs for the right hand side of the `ddove` equation (this list is derived from the arrows pointing into the `ddove` valve) and gives the user the opportunity to enter his or her code.

The full STELLA code derived from the diagram of Figure 3.2 is the following:

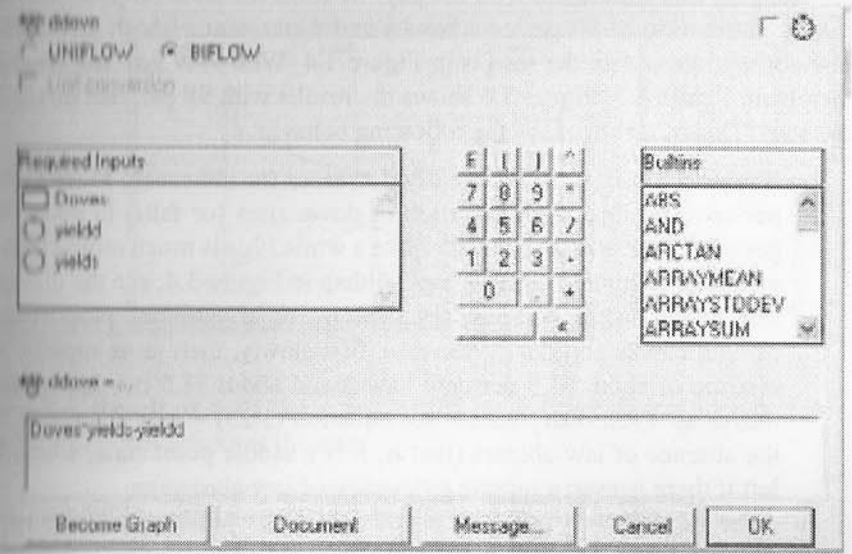
[2] A 'sink' in system dynamics terminology is a never-overflowing basin to which flows may be directed that leave the system; thus it is the opposite of a 'source'. Note that in linear stability analysis (see Appendix B) 'sink' and 'source' have a different meaning, namely stable and unstable stationary state, respectively.

Figure 3.2: System dynamics diagram of the dove–hawk–law-abider model




```
Doves(t) = Doves(t - dt) + (- ddove) * dt

INIT Doves = (1-InitialHawks)/2
ddove = Doves*yields-yieldd

Hawks(t) = Hawks(t - dt) + (ddove - dlawa) * dt

INIT Hawks = InitialHawks
ddove = Doves*yields-yieldd
dlawa = yieldl-LawAbiders*yields

LawAbiders(t) = LawAbiders(t - dt) + (dlawa) * dt
INIT LawAbiders = (1-InitialHawks)/2
dlawa = yieldl-LawAbiders*yields
```


Figure 3.3: Entering flow equations in STELLA




```
codo = 3
coha = 20
InitialHawks = 0.9
poss = 10
rdd = poss/2-codo
rdh = 0
rdl = rdd/2
rhd = poss
rhh = (poss-coha)/2
rhl = (rhh+poss)/2
rld = (rdd+poss)/2
rlh = rhh/2
rll = poss/2
yieldd = (Doves*rdd+Hawks*rdh+LawAbiders*rdl)*Doves
yieldh = (Doves*rhd+Hawks*rhh+LawAbiders*rhl)*Hawks
yieldl = (Doves*rld+Hawks*rlh+LawAbiders*rll)*LawAbiders
yields = yieldd+yieldh+yieldl
```

yieldd, ..., yields are converters (in DYNAMO: auxiliaries) which are used as shorthand for a longish expression such as (Doves*rdd + Hawks*rdh + LawAbiders*rdl)*Doves which could have replaced yieldd in line 3 of the above program code (but then with any change of this expression, one would have needed to change it several times).

Running this simulation with the payoffs from the table on p. 34 and an initial distribution of 90 per cent hawks and 5 per cent of both doves and law-abiders, we obtain the results in Figure 3.4. With 99.9 per cent hawks, we obtain Figure 3.5. Figure 3.6 shows the results with 99 per cent doves at the start. This model displays the following behaviour:

- The proportion of hawks rapidly decreases (or increases) to about 61 per cent, whereas the proportion of doves rises (or falls) to about 38 per cent. This level persists for quite a while (this is much more clearly visible in Figure 3.5 and Figure 3.6 than in Figure 3.4; see the discussion below, in the commentary section). Afterwards the proportions of both hawks and doves decrease, first slowly, then more rapidly. A mixture of about 61.5 per cent hawks and about 38.5 per cent doves makes up a stationary state – see equation (3.23) – which is stable in the absence of law-abiders (that is, it is a saddle point state, which is left if there is even a minute proportion of law-abiders).
- After the stationary state, the proportion of law-abiders increases very slowly.
- Later on, the proportions of both hawks and doves decrease (and eventually they become extinct), while the proportion of law-abiders rises to 100 per cent.

Of course, any population with only one strategy extant is at a stationary state. With the parameter values as applied above, only the last mentioned state – the extinction of hawks and doves – is a stable state. Even if the simulation starts with 99 per cent doves and 0.5 per cent of hawks and law-abiders each, only the latter survive (see Figure 3.6).

For Hobbes' theory we have two consequences:

- As soon as the law-abiding strategy, which is superior to the other two, was invented, it would necessarily prevail, and it would so by nature, not by covenant and only because of the individuals' capacity to inherit, imitate or learn.
- The law-abiding strategy prevails only after a considerable time. The time it takes until it first grows is the longer, the larger the initial proportion of hawks. The eventual success is rather sudden, the more so, the larger the initial proportion of hawks (compare Figure 3.4 to Figure 3.5).

Figure 3.4: Result of a STELLA run of the dove–hawk–law-abider model with 90 per cent hawks at the start

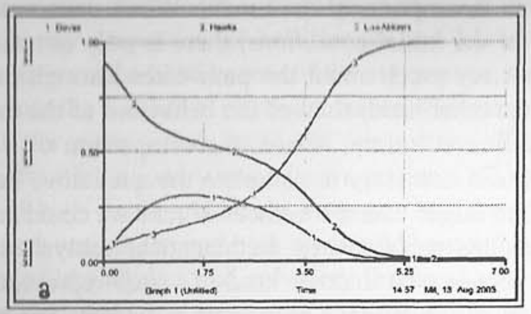


Figure 3.5: Result of a STELLA run of the dove–hawk–law-abider model with 99.9 per cent hawks at the start

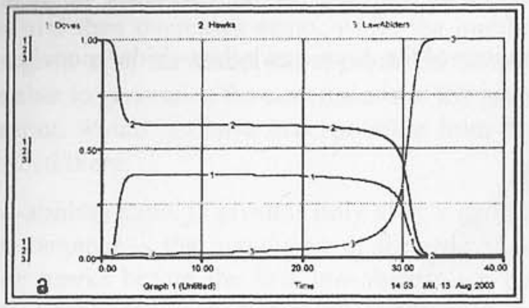
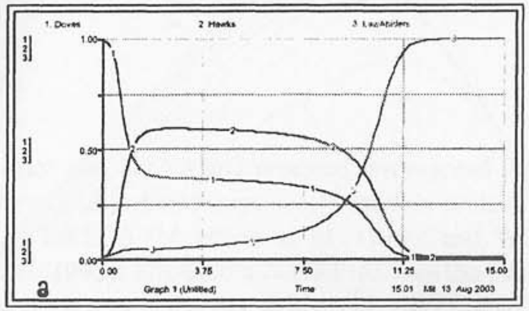


Figure 3.6: Result of a STELLA run of the dove–hawk–law-abider model with 99 per cent doves at the start

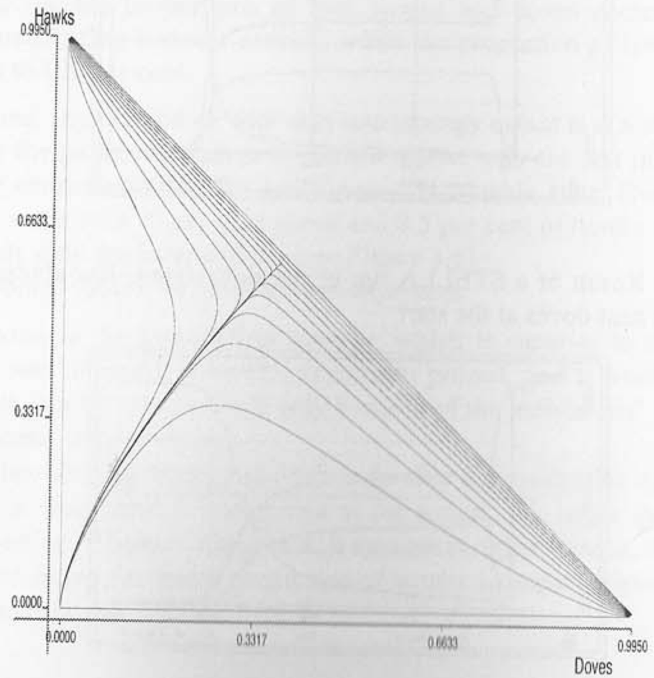


Commentary

Neither the mathematical treatment nor the simulation allowed a convincing qualitative overall description of the model. While mathematics taught us that, regardless of the initial conditions, there is only one stable state, the equations did not say much about the path taken through the state space. Simulation, on the other hand, showed the behaviour of the model, but only for one initialization at a time. Hence, the comparison of a large number of simulation runs is necessary to complete the qualitative description of a model's behaviour, larger than the number of runs we could present here.

To overcome this gap between mathematical analysis and single-run simulation, we choose next another kind of visual representation, namely the representation of the model's behaviour in its state space. For this we draw 20 of the paths the model takes through its state space (see Figure 3.7 – the state space is spanned by the proportions of doves and hawks, and every point on one of the curves represent a state explicitly defined by the proportions of doves and hawks; the representative point of a population

Figure 3.7: Behaviour of the dove–hawk–law-abider model in its state space



moves towards the origin, which in turn represents a state with no doves and no hawks, but only law-abiders). Note that this diagram does not indicate the speed with which the model changes state.

When the model starts with a large proportion of doves (and consequently with a tiny proportion of both hawks and law-abiders), that is to say, from the lower right-hand corner of the state space, first the number of hawks rises while the number of doves decreases. The number of law-abiders remains small for quite a time, until the proportions of doves and hawks approach the fourth stationary state (see equation (3.23)). From then on, the numbers of both doves and hawks decrease, and the proportion of law-abiders increases until in the end both doves and hawks die out. If we have a larger number of law-abiders and only very few hawks from the beginning, that is, if we start from the middle of the bottom of the state space, the number of hawks initially increases only slightly and afterwards decreases, while the number of doves decreases from the very beginning. If we start with many hawks and few doves and law-abiders (top left-hand corner of the state space), then the number of hawks decreases fast, the number of doves first increases and then decreases again, while the number of law-abiders only begins to grow after the model has approached the saddle point.

So we are able to generalize the conclusions of the previous section, and this generalization would not have been possible from the few simulation runs we described there:

- The law-abiding strategy prevails only after a considerable time. The more homogeneous the population at the start (a large majority of doves *or* hawks before the first law-abiders are born), the later its success, and the more sudden its rise (start from the bottom right-hand corner of the state space).
- If the first law-abiders are born into a mixed society of doves and hawks, they begin to multiply very soon (start from the saddle point).

World models

System dynamics and DYNAMO received widespread interest mainly because they were used to build large world models such as WORLD2 (Forrester 1971); WORLD3 (Meadows *et al.* 1974); and WORLD3 revisited (Meadows *et al.* 1992). Forrester's WORLD2 was the first and simplest of these. We will use it now to discuss some problems of large system dynamics models.

Figure 3.8: Main features of Forrester's world model

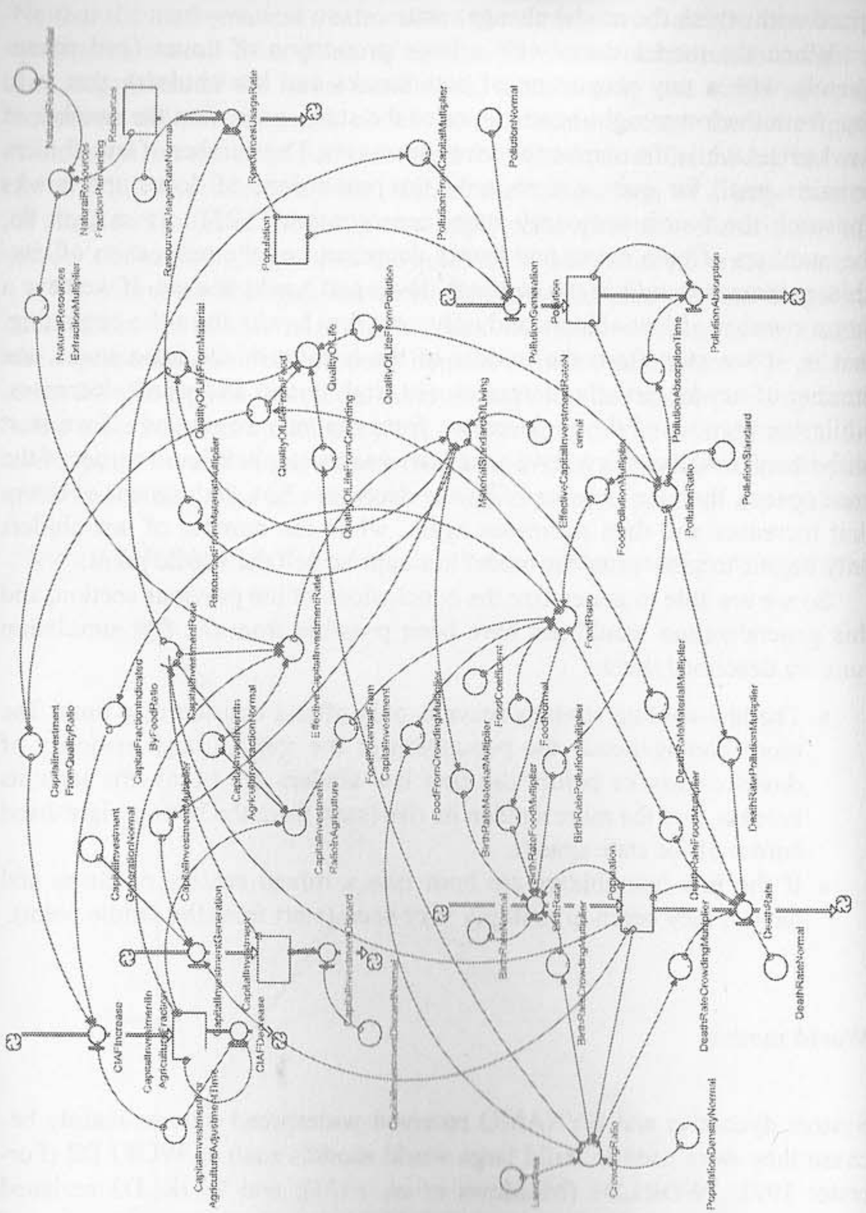


Figure 3.8 shows a STELLA version of Forrester's world model with its population sector, pollution sector, natural resources sector and capital stock sector. All these sectors contain one or two internal feedback loops. They are tied together by numerous auxiliaries and controlled by numerous constants.

The bottom part of Figure 3.8 shows some of the feedback mechanisms between the population and the pollution sectors. The corresponding lines of Forrester's program are shown below in a STELLA version:

- The population increases and decreases according to the birth and death rates:

```
Population(t) = Population(t - dt) + (BirthRate -
                DeathRate) * dt
```

- The birth rate depends on the actual population size, on a constant 'normal birth rate' and on several auxiliaries ('birth rate multipliers') for food supply, material life standard, crowding and pollution:

```
BirthRate = Population*BirthRateNormal*
            BirthRateFoodMultiplier*
            BirthRateMaterialMultiplier*
            BirthRateMaterialMultiplier*
            BirthRateCrowdingMultiplier*
            BirthRatePollutionMultiplier
BirthRateCrowdingMultiplier = GRAPH(CrowdingRatio)
(0.00, 1.05), (1.00, 1.00), (2.00, 0.9), (3.00, 0.7),
(4.00, 0.6), (5.00, 0.55)
CrowdingRatio = Population/(LandArea*
                PopulationDensityNormal)
LandArea = 135E6
PopulationDensityNormal = 26.5
BirthRatePollutionMultiplier = GRAPH(PollutionRatio)
  (0.00, 1.02), (10.0, 0.9), (20.0, 0.7), (30.0, 0.4),
  (40.0, 0.25), (50.0, 0.15), (60.0, 0.1)
```

The latter two (`BirthRateCrowdingMultiplier` and `BirthRate-PollutionMultiplier`) are determined by so-called table functions (see below). `BirthRateCrowdingMultiplier` and `BirthRatePollutionMultiplier` depend on `CrowdingRatio` (crowding) and `PollutionRatio` (pollution rate), respectively. `CrowdingRatio` is defined as proportional to the actual population size (for the latter, see below).

- The death rate also depends on the actual population size, on a constant death rate and, like the birth rate, on multipliers for food supply, material life standard, crowding and pollution:

```
DeathRate = Population*DeathRateNormal*
            DeathRateMaterialMultiplier*
            DeathRatePollutionMultiplier*
            DeathRateFoodMultiplier*
            DeathRateCrowdingMultiplier
DeathRateFoodMultiplier = GRAPH(FoodRatio)
   (0.00, 30.0), (0.25, 3.00), (0.5, 2.00), (0.75, 1.40),
   (1.00, 1.00), (1.25, 0.7), (1.50, 0.6), (1.75, 0.5),
   (2.00, 0.5)
DeathRateMaterialMultiplier =
   GRAPH(MaterialStandardOfLiving)
   (0.00, 1.80), (0.5, 1.80), (1.00, 1.00), (1.50, 0.8),
   (2.00, 0.7), (2.50, 0.6), (3.00, 0.53), (3.50, 0.5),
   (4.00, 0.5), (4.50, 0.5), (5.00, 0.5)
DeathRateCrowdingMultiplier = GRAPH(CrowdingRatio)
   (0.00, 0.9), (1.00, 1.00), (2.00, 1.20), (3.00, 1.50),
   (4.00, 1.90), (5.00, 3.00)
DeathRatePollutionMultiplier = GRAPH(PollutionRatio)
   (0.00, 0.92), (10.0, 1.30), (20.0, 2.00), (30.0, 3.20),
   (40.0, 4.80), (50.0, 6.80), (60.0, 9.20)
```

Again, the death rate multipliers (`DeathRateFoodMultiplier`, `DeathRateMaterialMultiplier`, `DeathRateCrowdingMultiplier` and `DeathRatePollutionMultiplier`) are determined by table functions different from the ones used for the calculation of birth rate multipliers.

- The pollution rate is calculated from the actual pollution level by a simple division:

```
PollutionRatio = Pollution/PollutionStandard
PollutionStandard = 3.6e9
```

- The level of pollution is determined by the rates of its generation and absorption:

```
Pollution(t) = Pollution(t - dt) + (PollutionGeneration -
               PollutionAbsorption) * dt
```


- Pollution generation depends on the population size, on a switchable constant, and on `polcm`, the 'pollution capital multiplier' determined by the capital sector, which we will not discuss here:

```
PollutionGeneration = Population*PollutionNormal*
                      PollutionFromCapitalMultiplier
```

- Pollution absorption depends only on the actual level of pollution, but in so intricate a manner that a table function is again used:

```
PollutionAbsorption = Pollution/PollutionAbsorptionTime
PollutionAbsorptionTime = GRAPH(PollutionRatio)
   (0.00, 0.6), (10.0, 2.50), (20.0, 5.00), (30.0, 8.00),
   (40.0, 11.5), (50.0, 15.5), (60.0, 20.0)
```

Table functions were DYNAMO's (and graph functions are STELLA's) means of modelling those nonlinear relationships between two variables that cannot be written down as a single equation. In most cases, these nonlinear relationships are taken from empirical data. In STELLA, function tables are defined with the help of a special window which is shown in Figure 3.9.

The value that `table` returns is calculated as a linear interpolation. The

Figure 3.9: Evaluation of table functions in STELLA

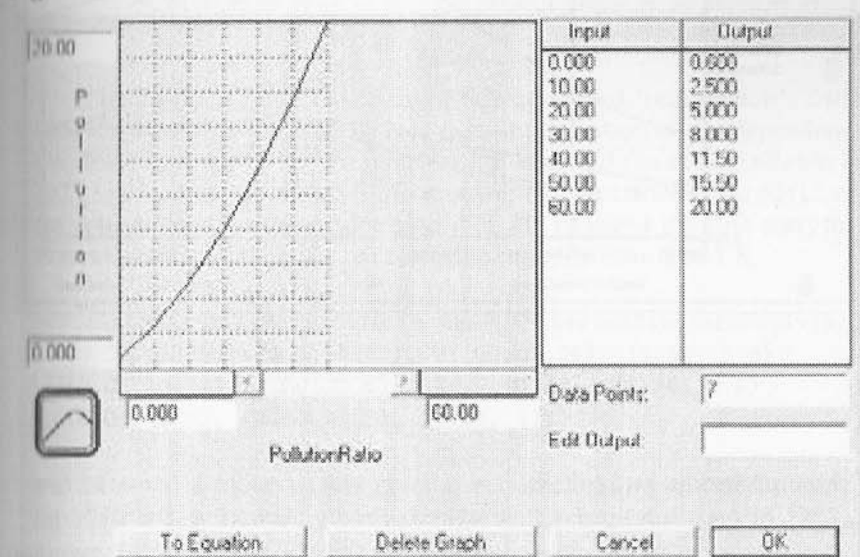
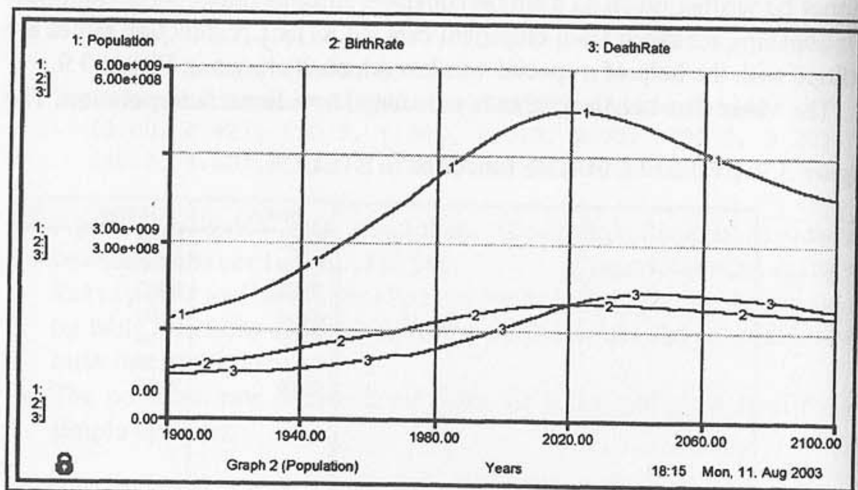


table function used in the calculation of pollution absorption is evaluated as indicated by Figure 3.9: function values when its argument is within the first interval are interpolated between the first and second table entries, function values when its argument is within the second interval are interpolated between the second and third table entries, and so on. Thus, the table must have $n + 1$ entries for n intervals.

The table function technique makes a large number of numerical values necessary in a STELLA program. With WORLD2's 22 table functions, this amounts to 151 numerical values.

Figure 3.10 shows its predictions for births, deaths and world population size. The latter is predicted to have its maximum about the year 2035 when, for the first time since the early twentieth century, the number of deaths will exceed the number of births.

Figure 3.10: Prediction results of Forrester's WORLD2 model for births, deaths and population size

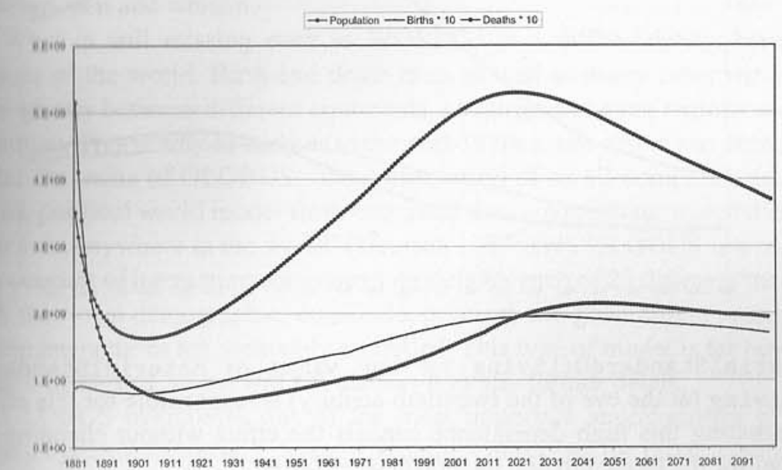


Problems and an outlook

It is interesting to see what happens when Forrester's world model, with its standard parameter set, is used to 'retrodict' births, deaths and world

population backwards in time (see Figure 3.11). We see immediately that there is a problem, because during the last two decades of the nineteenth century the world population is 'predicted' to have decreased from 6 billion in 1880 to the historical 1.7 billion in 1900, which was obviously not the case.

Figure 3.11: Retrodiction of Forrester's WORLD2 model back to 1880



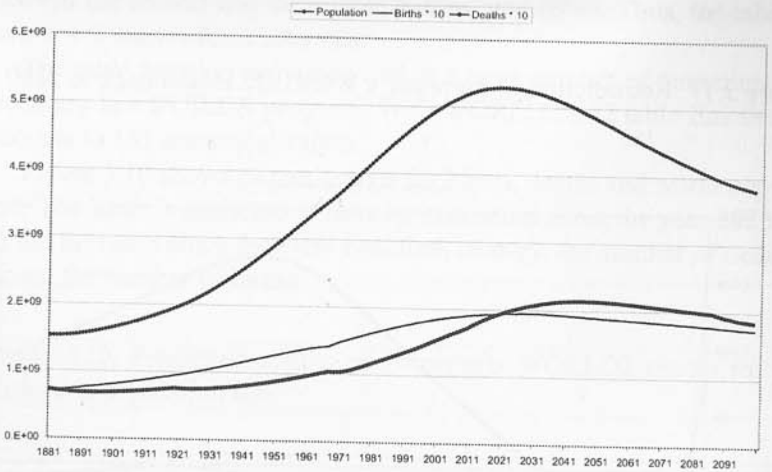
It is difficult to find the cause of this erroneous 'retrodiction'. Zwicker (1981: 481) points out that with a slight modification of the dependence of the death rate multiplier for material life standard (`DeathRateMaterialMultiplier`) on the material life standard (`MaterialStandardOfLiving`) the retrodiction is much more plausible. He changed the first entry in the `DeathRateMaterialMultiplier` table function from 3 to 1.8,

```
DeathRateMaterialMultiplier = GRAPH(MaterialStandardOfLiving)
   (0.00, 1.80), (0.5, 1.80), (1.00, 1.00), (1.50, 0.8),
   (2.00, 0.7), (2.50, 0.6), (3.00, 0.53), (3.50, 0.5),
   (4.00, 0.5), (4.50, 0.5), (5.00, 0.5)
```

and obtained a more or less correct 'retrodiction' of the total population for 1880 and, moreover, a birth rate above the death rate back to 1888 (see Figure 3.12).

The high dependence of `DeathRateMaterialMultiplier` on

Figure 3.12: Retrodiction of Forrester's WORLD2 model back to 1880, with a slight correction of `DeathRateMaterialMultiplier`



`MaterialStandardOfLiving` for low values of `MaterialStandard-OfLiving` (at the eve of the twentieth century) is responsible for this effect. Eliminating this high dependence cancels the effect without changing the model for the twentieth and twenty-first centuries.

Table functions can thus be dangerous – we should not forget that a table function is a fairly raw means of representing the dependence of one variable on another. In many cases, modellers have only a rough notion of this dependence and a notion such as 'the more of x, the faster y increases' may be represented by an infinite number of different continuous or table functions. Hence, modellers may fall into the 'trap of tractability' (Doran and Gilbert 1994: 13) when they select their representation of a monotonic dependence: a linear dependence is always the simplest form of a monotonic dependence, and it is easily tractable by mathematical algorithms, but solutions will usually be different for linear dependencies as compared with different nonlinear ones. This was one of the reasons for introducing so-called 'qualitative differential equations' (Kuipers 1994: 3) into the modelling and simulation scene. The only type of knowledge used in qualitative simulation is in terms of intervals between 'landmarks' – for example, the interval between the melting point of ice and the boiling point of water – and in terms of monotonically increasing and decreasing functions. Qualitative simulation has so far mostly been applied to physical phenomena ('naïve physics'), and only seldom to social phenomena (but see Brajnik and Lines 1998), so we will not go into the details of this new approach.

Another shortcoming of Forrester's WORLD2 is the fact that the population is always seen as a whole and that its age structure is not considered at all. A changing age structure, however, will affect both birth and death rates. Thus, Meadows' WORLD3 was a step forward in so far as this world model contained four different level variables for the age groups 0–14, 15–44, 45–64 and 65+, with different death rates and birth rates depending only on the population aged between 15 and 44. However, the model did not distinguish between men and women.

What is still missing even in WORLD3 is a differentiation between regions of the world. Birth and death rates as well as many other variables vary greatly between different continents, countries and even regions within countries. This is why as early as in the mid-1970s a new effort was launched under the name of GLOBUS: 'the construction of an all-computer, nation-based, political world model from empirical data – something which did not then exist anywhere in the world' (Deutsch 1987: xiv). GLOBUS is a model that consists of interacting component models for each of 25 different nations with their own demographic, economic, political and government processes whose interactions are separately modelled. This type of model is far beyond system dynamics, so we will not discuss it in any further detail.

GLOBUS overcomes one of the most important shortcomings of the system dynamics approach. System dynamics describes the target system as a single entity or object. A system dynamics model is an indivisible whole. If we happen to find parts in the target system (like continents or countries in the world) we have to describe their properties as attributes of the world model and thus leave the system dynamics approach – as GLOBUS did.

Although the GLOBUS group never continued their research after their book appeared and after MicroGLOBUS (a DOS-based demonstration model) had been distributed, there are other groups who followed similar approaches. The 'International Futures' Group (Hughes 1999) developed a model encompassing all major states of the world (which can be aggregated arbitrarily into regions.[3] Population is modelled in five-year cohorts, several economic sectors, food types, land types, energy types and types of government spending can be distinguished. Thus the International Futures model is, of course, much more detailed than the classical world models by Forrester and Meadows, and even more detailed than the GLOBUS model in so far

[3] The downloadable demonstration and student version, `http://www.du.edu/~bhughes/ifswelcome.html`, comes with nine individual countries, the European Union and seven regions such as 'Other Europe'.

as the latter encompasses only 25 nations plus the 'rest of the world', has a coarser-grained age structure, which is exogenously determined, just to name a few differences.

Further reading

There are many books dedicated to the system dynamics simulation approach, beginning with

- Forrester, J. W. (1980): *Principles of Systems*, 2nd preliminary edn. MIT Press, Cambridge, MA (1st edn 1968).

which first introduced the technique. It includes a number of technical details about an early version of DYNAMO and some simple examples. This technique was first applied by

- Forrester, J. W. (1971) *World Dynamics*. MIT Press, Cambridge, MA

to world models of the type we discussed earlier in this chapter, and

- Forrester, J. W. (1969) *Urban Dynamics*. MIT Press, Cambridge, MA

applied system dynamics to 'the problems of our ageing urban areas', introduced a model of an urban area and predicted over 250 years its future development in the unemployment, labour, managerial and professional sectors as well as in the housing, industry, tax and town planning sectors. Forrester's first book on related topics,

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

has enjoyed a wide readership and stimulated research on complex systems. The original DYNAMO manual was

- Pugh, A. L. III (1976) *DYNAMO User's Manual*. MIT Press, Cambridge, MA

which has since been superseded by more modern versions of the DYNAMO language.

Another influential group of books began with the introduction of a far more sophisticated world model in

- Meadows, D. L. *et al.* (1974) *Dynamics of Growth in a Finite World*. MIT Press, Cambridge, MA.

This described the world population in different age groups, distinguished between industrial and service capital, and went into more detail concerning land use and fertility. Its results are discussed from a 1990s perspective in

- Meadows, D. H. *et al.* (1992) *Beyond the Limits*. Chelsea Green, Post Mills, VT.

This book states that the original model needed only very few corrections, after the data produced by the target system – the world as it behaved in the 1970s and 1980s – were taken into consideration.

A comprehensive description of system dynamics oriented simulation methods in the social sciences is provided by

- Hanneman, R. A. (1988) *Computer-Assisted Theory Building. Modeling Dynamic Social Systems*. Sage, Newbury Park, CA.

He does not so much address a special target system (like Forrester and Meadows always did, writing about urban or industrial or world development), but rather has 'the immodest goal of reorienting how many social scientists go about building and working with theories' (p. 9), thus making simulation a new methodological paradigm for the social sciences, restricting himself, however, to macro and other equation-based models throughout the book.

An extensively comprehensive description of system dynamics oriented simulation mostly, but not only, in business research was recently published as

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

It comes with a CD-ROM with modelling software from Vensim, ithink and PowerSim dedicated to 'issues such as fluctuating sales, market growth and stagnation, the reliability of forecasts and the rationality of business decision making.' (from the blurb)

More recent world models are presented and discussed in

- Bremer, St.A. ed. (1987) *The GLOBUS Model. Computer Simulation of Worldwide Political and Economic Developments*, Campus/Westview Press, Frankfurt/M. and Boulder, CO.

which is the summary of work done in the GLOBUS project which developed a world model 'based on nation-states, not regions' while

- Hughes, B.B. (1999) *International Futures: Choices in the Face of Uncertainty*, Westview Press, Boulder, CO.

in a way continues this work in so far as it presents a more modern (Windows compatible) type of multi-nation world model which can be downloaded from `http://www.du.edu/~bhughes/ifswelcome.html`.

The qualitative simulation approach briefly mentioned on page 52 is discussed in detail in

- Kuipers, B. (1974) *Qualitative Reasoning. Modeling and Simulation with Incomplete Knowledge*, MIT Press, Cambridge, MA.

Chapter 4

Microanalytical simulation models

As discussed in the previous chapter, system dynamics models its target systems as indivisible wholes and does not take into account the fact that for the social scientist target systems usually consist of individual persons, groups, classes, subpopulations and so on. Social scientists will therefore be interested in modelling approaches on several levels – an aggregate level and at least one lower level. The first approach that tried to solve this problem was the classical microsimulation approach. It has been used to predict the individual and group effects of aggregate political measures that often apply differently to different persons. For instance, a tax formula that imposes taxes only on persons with incomes above a certain threshold might be changed by moving this threshold. If we want to calculate the gross effect on the total tax revenue, a simulation on the macro level cannot help. We must instead go back to the individual cases, calculate their taxes due before and after the tax revision, and reaggregate the tax revenue.

Another example can be taken from demography. Changing age structures of a population can be simulated on a macro level – see the discussion on page 53 in Chapter 3. We would have several level variables with the sizes of a number of sex/age groups to which we would apply age-group-specific death rates, and we would calculate births from the sizes of the female age groups between 15 and 45 years of age, applying age-dependent fertility rates. If we were only interested in the age structure of a population, this deterministic macro model might be sufficient because, with a very large