The Full Potential of Continuous System Simulation Modelling

Leif Gustafsson, Mikael Sternad, Erik Gustafsson

Signals and Systems, Department of Engineering and Sciences, Uppsala University, Uppsala, Sweden
Email: Leif.Gunnar.Gustafsson@gmail.com, Mikael.Sternad@signal.uu.se, Ekir.Gustafsson@gmail.com

Abstract

Continuous System Simulation (CSS) is a powerful way to study the behaviour of differential-algebraic equation models. Differential-algebraic equation modelling goes back to Newton and works well for models of e.g. physical systems where stochasticity plays almost no role, and where only a single attribute of an object is studied over time. However, when results from a deterministic CSS model are compared with results from a Discrete Event Simulation (DES) model, they are often inconsistent. The reasons behind this CSS-DES inconsistency are nowadays well understood. In this paper, we demonstrate that a CSS model can contain both continuous state variables (compartments) that change continuously over time and discrete state variables (also compartments) that model discrete entities and change by integer amounts. In both cases, the time-slicing method is used to advance time. Furthermore, stochasticity can and should play the same role in CSS as it does in DES. This paper first explains how a well-defined conceptual model can be stepwise transformed in a consistent way into a CSS model. These transformation steps provide insights into how to construct a consistent CSS model. In short, this is about preserving uncertainties, attributes and dynamic properties. An additional benefit of this approach is that continuous and discrete sub-models can interact within the same model, without having to combine different types of simulation languages, types of time handling and incongruent concepts. To facilitate the use and understanding of stochastic CSS, this paper is also intended to serve as a guide to performing consistent CSS modelling and simulation. Furthermore, an open source tool for collecting and analyzing the outputs from a stochastic CSS model and analyzing and presenting the results in statistical form is also developed and made available to the reader.

Keywords

Continuous Modelling, CSS, Discrete Modelling, DES, Combined Simulation,

Compartment-Based Model, Stochastic Modelling

1. Introduction

When Continuous System Simulation (CSS) and Discrete Event Simulation (DES) became established methods in the 1960s, it was soon noted that the results from these modelling methods often differed. The reasons behind this inconsistency and the problem of what type of model to use under different circumstances have been discussed since then [1]-[6].

In a series of papers [7]-[12], we have compared different simulation methods and analysed conditions for these methods to produce consistent results. In these studies we have also analysed and compared the model size and complexity when using these methods under different conditions. In this paper, we demonstrated how CSS can be constructed in a way that eliminates all types of bias and artifacts so that CSS and DES produce results that are fully consistent. In brief, the underlying problems originate from the conventional way of performing CSS modelling. By taking account of the following four fundamental issues that have been neglected in the past, full consistency can be obtained:

1) Discrete quantities and discrete information-bearing signals should usually be modelled as discrete, while continuous matter and continuous information-bearing signals should be modelled as continuous.

2) Relevant properties of real objects have to be considered as model attributes.

3) The dynamic properties of a stage have to be preserved. (The concepts of stage and compartment should not be confused. A stage usually has to be modelled by a structure of compartments.)

4) Uncertainties of different kinds must be correctly represented in the model, often leading to a stochastic model.

Based on the papers [7]-[12], a comprehensive theory for how this is performed is presented in this paper. All four issues can be handled smoothly within the CSS concept, as is demonstrated in this paper. Furthermore, a stochastic model generates stochastic output. Therefore, a device to obtain statistical results from multiple simulations is required. For this reason, we provide an open source tool (embedded in a simple CSS language) for collecting and analyzing the outputs from a stochastic CSS model and analyzing and presenting the results in statistical form.

The purpose of this paper is to demonstrate how to take the four aspects listed above into account, in order to give CSS modelling its full potential while maintaining the consistency with a given conceptual model. The paper is also intended to serve as a guide to performing consistent CSS modelling and simulation.

Although a CSS model can include both continuous and discrete quantities,

we focus more on the latter because the former can be assumed to be well known.

In Section 2, we introduce how continuous and discrete quantities can be handled in a CSS model and outline some problems with continuous modelling of discrete objects. In Section 3, we show how a well-defined conceptual model can be stepwise transformed into a consistent CSS model. Appropriate handling of the attributes and the dynamic properties of a stage then appears as a natural consequence. In Sections 4 and 5, we discuss these aspects further and show the consequences of neglecting correct handling of attributes and of confusing stages with compartments. In Section 6, we show that both continuous and discrete quantities can be combined within the same CSS model, while in Section 7, we discuss the different types of uncertainties and how these can be implemented in a CSS model, often as stochasticities. A stochastic model produces stochastic variations both within a replication (simulation run) and between replications that usually have to be analysed in statistical terms, an issue discussed in Section 8. Then discussion and conclusions follow in Section 9: we also include Appendix A containing six illustrative examples that are referred to from many points in the text. These models can be downloaded and explored using the software referred to in Appendix B.

2. Continuous and Discrete Modelling in CSS

Conventional CSS usually treats all compartments (state variables) as continuous, and the time handling is based on time-slicing that transfers fractions of the content in a compartment during a time-step. However, in many situations, some quantities of interest to be modelled are, by their nature, discrete. There is then nothing to prevent us from modelling the contents in compartments as integers (of *discrete tokens*) that transition the model [7] [13]. Preserving discreteness and continuousness in the model will remove bias and artifacts of different kinds. The practice of modelling everything as continuous in CSS is a main reason for inconsistent behaviours and results.

In the *continuous case*, a fraction of the continuous matter in a compartment is transferred during a small time-step. The transitions to/from/between compartments are often deterministic, but may vary stochastically because of unpredictable changes in the environment (parameter stochasticity, see Section 7.6).

In the *discrete case*, discrete quantities (tokens) are transferred to/from/between the compartments during a time-step. The transitions can be deterministic, e.g. described by a pulse train or according to a time table, or observed behaviour described by a table look-up function. However, the transitions often have to be described in stochastic terms. The uncertainties in the discrete case may originate from knowing only the average rate of a transition (transition stochasticity to be treated in Section 7.4) and from unpredictable changes in the environment (parameter stochasticity, see Section 7.6) that affect the average rate.

2.1. Continuous Modelling

Continuous modelling is closely related to differential-algebraic equation models. A continuous (matter) model can be constructed from observations or translated from known differential and algebraic equations. If the differential equation is of the *n*th order, it can always be rewritten as *n* coupled first-order equations, each describing the rate of change of one state variable (compartment).

Below a continuous first-order difference equation (written in Euler's form) based on the compartment X is initiated and updated by inflows and outflows for each (sufficiently small) time-step Δt by:

$$X(0) = \text{rational number} \quad \text{(called } \textit{real} \text{ in computer jargon)} \tag{1a}$$

$$X(t + \Delta t) = X(t) + \Delta t \cdot F_{in}(t) - \Delta t \cdot F_{out}(t) \tag{1b}$$

$$\Delta t \cdot F_{in}(t) = \Delta t \cdot \lambda(t) \quad \text{(where } \lambda \text{ is the inflow intensity)} \tag{1c}$$

$$\Delta t \cdot F_{out}(t) = \Delta t \cdot \mu(t) \quad \text{(where } \mu \text{ is the outflow intensity)} \tag{1d}$$

In a programme, this would be coded: $F_{in}(t) = \lambda(t)$ and $F_{out}(t) = \mu(t)$, since the product, $\Delta t \cdot F$, is not allowed on the left-hand side of an assignment statement. The form in (1c) and (1d) is used here to display the similarity with the discrete case below.

2.2. Discrete Modelling

In a discrete model or sub-model, the compartments must be initiated by integer values (of tokens), and the transitions must preserve the discreteness.

To initiate a discrete number of tokens, just let the content of each compartment be an integer:

$$X(0) = \text{integer number} \tag{2a}$$

Because discreteness cannot be preserved by transferring fractions, it becomes necessary to either transfer zero or an integer number of tokens at each Δt.

The transitions can be deterministic or stochastic. In the deterministic case, the difference equation model (1) applies with the extra condition that $\Delta t \cdot F_{in}(t)$ and $\Delta t \cdot F_{out}(t)$ must be integers. For example, transitions may follow a regular pattern or be described by a table look-up function.

However, in most cases the detailed information to model the instants when the tokens will transit is not available. For example, there is no exact information on when an atom will decay, an individual will cross the street, a customer will arrive etc. Then it is only possible to describe the probability distribution of the number of transitions per time unit and draw integer random numbers from this distribution, which makes the model stochastic.

Furthermore, if the intensities are random and independent, we have a Poisson process. Then the number of transitions during a time period (here Δt) becomes Poisson distributed, symbolically denoted $Po[\Delta t \cdot \lambda]$ and $Po[\Delta t \cdot \mu]$, with expected values $\Delta t \cdot \lambda$ and $\Delta t \cdot \mu$, respectively [14]. The Poisson process

may also be time varying. However, during a small time-step the intensities, $\lambda(t)$ and $\mu(t)$, can be regarded as constant, see [7].

In the stochastic case, the state Equation (2b) of a discrete model is then the same as for the continuous Equation (1b), but the flow equations get the Po[ ] clause:

$$X(t + \Delta t) = X(t) + \Delta t \cdot F_{in}(t) - \Delta t \cdot F_{out}(t) \tag{2b}$$

$$\Delta t \cdot F_{in}(t) = Po\big[\Delta t \cdot \lambda(t)\big] \tag{2c}$$

$$\Delta t \cdot F_{out}(t) = Po\big[\Delta t \cdot \mu(t)\big] \tag{2d}$$

In a programme, this is coded: $F_{in}(t) = Po\big[\Delta t \cdot \lambda(t)\big]\big/\Delta t$ and $F_{out}(t) = Po\big[\Delta t \cdot \mu(t)\big]\big/\Delta t$, since the product, $\Delta t \cdot F$, is not allowed on the left-hand side.

Handling the uncertainty of the number of transitions per time-step is called *transition stochasticity* and is further discussed in Section 7.4. (In addition, the intensities $\lambda(t)$ and $\mu(t)$ may also vary stochastically to describe uncertainties in the environment.)

Discrete and continuous processes can be freely combined in a CSS model (see Section 6 and the discrete/continuous Volterra model in Example A6 in Appendix A).

Equation (1) is said to be *embedded* in Equation (2). It is obtained by removing the Po[ ]-clauses in (2c) and (2d). The embedded continuous dynamic model (1) can be seen as an approximation of the discrete dynamic model (2), which may or may not distort the model behaviours. In the next subsection, we briefly discuss some consequences of describing discrete objects as continuous.

2.3. Consequences of Continuous Modelling of Discrete Objects

CSS is often used to model both continuous matter and discrete objects by a continuous equation system like (1) above.

However, modelling all physical or information-bearing quantities as continuous is not acceptable in most situations, because a discrete and a continuous process, ceteris paribus, are profoundly different and will usually produce different results and even show qualitatively different behaviours. Furthermore, the transitions of discrete objects are often irregular and dependent on so many unknown factors that they have to be described as stochastic. Transition stochasticity has no counterpart in a continuous flow. For a more complete discussion, see [11]. Deterministic and stochastic modelling approaches are also compared for specific studies in many papers, e.g. [15].

Below, we briefly mention some important aspects of replacing equation system (2) with its embedded equation system (1). This is also demonstrated by a number of recurrent examples collected in Appendix A[1].

- Some models, e.g. queuing models, require transition stochasticity to be

---

[1]For full understanding, it may be practical to study these examples after a full reading. The examples A1 to A6 are also available as executable models on the net, see Appendix B.

meaningful (see Example A5). For example, an M/M/1 queue with the intensity of 9 arrivals per time unit and a capacity to serve 10 individuals per time unit will produce long queues and waiting times in the discrete and stochastic setting, but erroneously no queues or waiting times in the deterministic setting.

- A continuous model may hide behaviours and models with very different model structures may falsely produce the same behaviour when discrete objects are approximated as continuous matter (see Example A2).

- A continuous model used to describe discrete objects prevents 'stochastic jumps' to solutions that are possible to reach in a discrete model for a given set of initial values. For example, in a continuous Volterra model, a species cannot become extinct (see Example A6) and in a continuous Lanchester's model of warfare, a weaker force can never defeat a stronger one (see Example 2 in [11]).

- A continuous model used to describe discrete objects can erroneously reach a stationary equilibrium, while the stochastic transitions between compartments of a discrete model may continually excite the model (see the Volterra model in Example A6 and the logistic model in Example A2-1).

- A continuous model used to describe discrete objects will in general produce results that differ from those of the expected values from a discrete model, e.g. because of nonlinearity in equations or output functions. For example, in a (bilinear) SIR model (see Example A4), the results may differ by many hundred percent. Even for a trivial linear model of radioactive decay, the time to the last decay is finite for the discrete model, but infinite for the embedded continuous model (see Example A1.) Also for the linear Lanchester model, the expected outcome and the time of the combat will also differ from the results of an embedded continuous model (see Example 2 in [11]).

- Even when a continuous model used to describe discrete objects produces unbiased estimates (see [11] for such cases), stochastic variations are removed or reduced. In the latter case, confidence intervals around estimates will be erroneously small because of the eliminated transition stochasticity.

Replacing the system of Equation (2) by its embedded system of Equation (1) entails two simplifications. One is replacing discreteness with continuousness and the other is removal of (possible) transition irregularities.

Experienced irregularity because of lack of exact and very detailed information is often unavoidable in modelling. We do not understand e.g. the processes in the nucleus causing the decay of a radioactive atom, when the next person will be infected by a disease or when the next traffic accident will occur and which vehicles will be involved etc. However, in some cases discrete objects behave in a pattern regular enough to be deterministically described. For example, new objects may be manufactured at regular intervals. In other cases, the rate of decay of radioactive atoms or the rate of H_2O molecules flowing in a river is so large that a continuous approximation is adequate. Transition stochasticity is then

almost removed because of the law of large numbers.

Whether a continuous simplification is appropriate may also depend on the question to be answered. For example, for a radioactive decay of N atoms with the time constant D, the (expected) time until the last decay is the same whether the discrete process is stochastic or deterministic (*i.e.* regular with $N(t)/D$ decays per time unit). However, if it is modelled as continuous, the time until the last decay will become infinite. So here it is the continuousness that generates the error. However, in e.g. the queuing example listed above, it is the removal of transition irregularities that erroneously eliminates the queue.

In special cases where the number of objects is so large that they can be regarded as a continuous matter and for many but not all linear systems, a discrete and an embedded continuous model will have the same, or almost the same, behaviour. The continuous model may then be preferred. In particular, parameter estimation and optimization are considerably easier to perform with a continuous model, see [11].

The conclusion from this is that discrete quantities should usually be modelled as discrete, while continuous matter should be modelled as continuous. Furthermore, it is important to describe the transition pattern, regular or irregular, in a proper way.

2.4. Finding a Proper Integration Method and Time-Step

2.4.1. Systems of Ordinary Differential Equations
1) *Integration methods*

For deterministic modelling of continuous matter, there exist a large number of integration routines for systems of ordinary differential equations that have different qualities. If the model develops smoothly, higher-order routines can produce very accurate results with a reasonable or even self-adjusting time-step. Because many CSS models include discontinuities requiring restarting of the algorithm, many CSS languages contain single-step routines, such as the Runge-Kutta routines of different orders that are fairly robust and fast. There is a vast body of literature on numerical integration methods for ordinary differential equations, e.g. [16] [17].

2) *Time-step*

After the choice of integration method, the size of the time-step, here denoted DT, has to be adjusted. Often DT = 1/6 or 1/10 of the shortest time-constant in the model is recommended (which of course also depends on the required accuracy of the result). The step-length is a compromise between accuracy and execution speed. A practical way to find an appropriate DT is to double or halve DT until a time-step is reached where the difference in results is negligible.

2.4.2. Systems in the Stochastic Case
1) *Integration methods*

High-order routines are not suitable for stochastic systems with discrete tokens, because such methods presuppose continuity in trajectories and their de-

rivatives. In particular, transition stochasticity creates discontinuities in each time-step. Although Euler's method with a proper DT works well, the issue of the best integration method for different types of stochastic models with discrete tokens has not been sufficiently studied. Gillespie discusses both Euler and second-order Runge-Kutta algorithms for stochastic simulation of chemical reaction systems [13]. A number of studies of efficient methods for stochastic simulation have been published since then, e.g. [18] [19] [20] [21].

2) *Time-step*

In the stochastic case, finding a proper DT is still more important than in the deterministic case because of the many replications that are always required to build probability distributions of the results. However, here it is also more complicated to find the proper DT.

A complicating difficulty with a stochastic model compared with a corresponding deterministic model is that DT cannot simply be adjusted until the difference in results becomes negligible, not even with seeds to every random number generator, because when DT is changed a different number of random numbers is used and the random numbers will not enter at the same places and points in time. Instead, it is necessary to make many replications with one DT and then with a time-step of half (or double) the size and to compare the generated distributions of the quantities of interest, or at least compare the averages (and e.g. variances) from these distributions.

It is our impression that for many stochastic models, Euler's method can be used with a time-step about the size (or half the size) of the time-step used in the corresponding deterministic and continuous model. However, this rule-of-thumb should be tested in each actual case.

3. Insights from Stepwise Transformation of a Conceptual Model into a CSS Model

Modelling (for analysis or simulation) means creating a simplified model in accordance with a specific and well-defined purpose in order to understand a system under study. It is "never" possible to obtain a true model. The best one can hope for is to obtain a model that truly realises a well-specified conceptual model (CM). We denote such a model *CM-consistent*, which means that an executable model should be fully consistent with a well-specified conceptual model.

CSS modelling is a more abstract representation than is usually noted in the literature and practice. To illustrate this, here we exemplify how a conceptual model (CM) that describes discrete entities can be stepwise transformed into a CSS model (a system of difference and algebraic equations) that is CM-consistent. These steps go via two types of DES models, called agent-based model (ABM) and entity-based model (EBM). Since this analysis is presented in detail in [12], in this section we keep the term "compartment-based model" (CBM) that was used there. When time is updated by time-slicing, CBM is the same as CSS.

To be more concrete, we specify a simple conceptual SIR model, *i.e.* an epidemic model based on the three *stages S* = Susceptible, *I* = Infectious and *R* = Recovered. (This type of epidemic model was first formulated by W. O. Kermack and A. G. McKendrick [22].) We then let the transformations demonstrate what is necessary for constructing a consistent CBM/CSS model.

Example 1: Realization of a conceptual SIR model as a consistent CBM/CSS model

First, we define a *conceptual SIR model* as an epidemic model of a population of 11 discrete **subjects**, of which six are males and five are females.

Attributes: Each subject has a unique *Identity* = {1, 2, ⋯, 11}, a *Sex* = {*Male*, *Female*}, and is in one of three *Stages* = {*S, I, R*}.

Process: Each of the subjects meets everybody else. A susceptible subject may then become infected by an infectious subject with a specified probability, *p*, per time unit. An infectious subject recovers after a sojourn time *D* in the *I*-stage (assumed to be *3*-Erlang(3/*D*) distributed) and thereafter becomes immune to the disease.

Finally, the **purpose** of the model study is: *To estimate the size and duration of a possible epidemic.*

In Figure 1, the conceptual SIR model (CM) is stepwise transformed into an ABM, EBM and CBM (*i.e.* CSS model).

The stepwise transformations T1-T5 from the conceptual model to a CM-consistent compartment-based model are discussed below:

Conceptual Model

The formal structure of the conceptual model is:
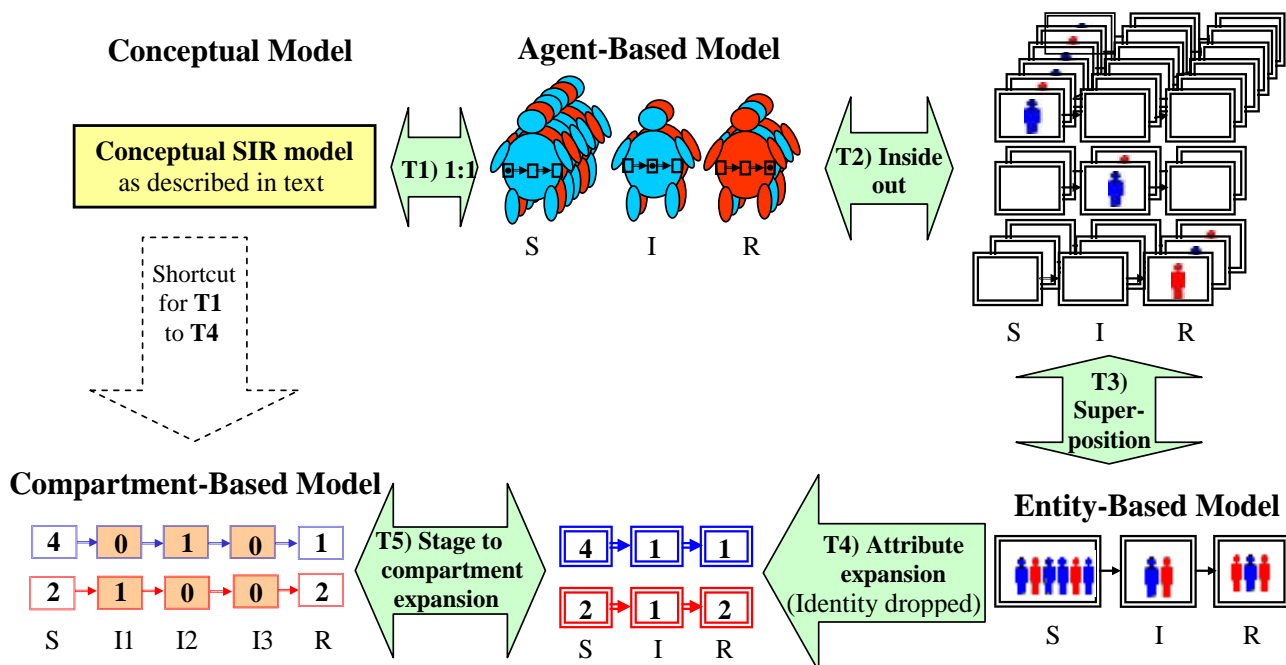
Figure 1. The stepwise transformations from conceptual model via ABM and EBM into a CBM/CSS model. (A stage is denoted by a double frame and a compartment by a single frame.)

Subjects{**Attributes**[Id, Sex, Stage], **Procedure** S → I(sjtd) → R};

where "sjtd" denotes the sojourn-time distribution. Hierarchically, the subject containing attributes and procedures is here the primary unit of the conceptual model.

T1: An agent-based model is a 1:1 mapping of the conceptual model in terms of *agents* with intrinsic, subordinated attributes and procedures.

Agent-based model

Because of the 1:1 mapping, the formal structure of the CM is preserved by the ABM:

Agents{**Attributes**[Id, Sex, Stage], **Procedure** S → I(sjtd) → R}.

Thus, the ABM is an executable version of the well-defined CM, so it is *CM-consistent*. The agents can interact with each other and with the environment.

T2: The transformation from an agent-based to an entity-based model is based on an *inside-out transformation*, where the intrinsic procedure of a former agent becomes superior and the agent is reduced to a subordinate entity (only containing attributes) that traverses the procedure. Figuratively, the entity traverses a flowchart of stages defined by the process.

T3: In the next transformation, all procedures (flowcharts) are *superposed* into a common overall procedure where the entities travel between stages. On arrival in a stage, attributes of the entity, other entities and global attributes may be changed. Then the entity will reside in the stage during a specified or random sojourn time:

Entity-based model

Procedure: S → I(sjtd) → R with {**Entities**[**Attribute**(Id, Sex, Stages)]}.

Further transforming an EBM into a CBM requires two modifications:

T4: The attributes connected to the entities must now instead be connected to the stages. This transformation, T4, is denoted *attribute expansion*. Taking the attribute "*sex*" in our example into account requires the expansion from three to six stages, namely S, I, R for males and S, I, R for females. Thus, a stage can only take the value of a single attribute combination (e.g. "*Disease stage*" and "S*ex*"), which means that every attribute taken into account will multiply the model size.

The stage could also preserve the attribute "*Identity*" of the entities, but that would make the model $N = 11$ times larger. Dropping the identity information makes the transformation T4 irreversible (denoted by a one-directional arrow), but it is still consistent (contradiction-free). In this example, the purpose is "*to estimate the size and duration of a possible epidemic*", so there is no need to preserve identity.

T5: The sojourn time in a stage may have any distribution, but the distribution for a single compartment is exponential here[2]. As discussed in Section 5, a stage with a given sojourn-time distribution can be approximated arbitrarily well

---

[2]When the *continuous* output rate from a compartment (X) is proportional ($1/D$) to its content, we get $dX/dt = -X/D$, resulting in an exponential sojourn-time distribution. In the *discrete and stochastic* case we get $dX/dt = -Poisson[dt \cdot X/D]/dt$, also producing an exponentially distributed sojourn time with expected value D.

by a linear system of compartments in series and/or parallel. This is called *stage-to-compartment expansion*. The *3*-Erlang (D/3) infectious I-stage can, however, be exactly represented by three serially connected compartments I1, I2 and I3 in a CBM [23].

Compartment-based model (*i.e.* a CSS model when time-slicing is used)

Procedure: $S_M \to I1_M \to I2_M \to I3_M \to R_M$ & $S_F \to I1_F \to I2_F \to I3_F \to R_F$;

where M stands for male and F for female.

In the CBM representation, only the procedure now remains. It has the form of a structure of compartments, where a compartment only contains a number of tokens. Attributes (*Sex*, *Disease stage* and possibly *Id*) and sojourn-time distributions are here structural parts of the procedure. Each stage then represents an *attribute combination*, and by replacing the stage by a number of compartments (state variables) in series and/or parallel, the prescribed sojourn-time distribution is approximately obtained. ■

This is only a simple example, but it provides the fundamental principles and insights for CSS modelling of discrete subjects. The principles are the same even when e.g. queues are involved or when position in space is an important attribute. This is further discussed in [12].

The approximation of a continuously varying attribute into discrete intervals and the approximation of a sojourn-time distribution of a stage can both be made sufficiently accurate, but at the expense of model size.

The shortcut, shown in **Figure 1**, of directly mapping the conceptual model into a CSS model is natural and efficient, *provided that the modeller understands how to perform it in a consistent way.*

In the following sections, we systematically discuss attribute expansion, stage-to-compartment expansion and how to include different types of uncertainties about the system under study. We also demonstrate the consequences of neglecting to perform these tasks properly.

4. Attribute Expansion

In agent-based and entity-based modelling, the cost of including an attribute is almost negligible, but for CSS modelling the cost grows combinationally with the number of values each attribute can take.

In Example 1 above, the conceptual model (besides the stage attribute) has the attributes *Identity* = {1, 2, ···, 11} and *Sex* = {*Male, Female*}. Because the purpose to estimate the size and duration of an epidemic does not require identity, this attribute could be dropped. Then the SIR model size was only doubled because of the attribute *Sex* = {*Male, Female*}. However, if the conceptual model were to require e.g. the attributes *Identity* = {1, 2, ···, 11}, *Sex* = {*males, females*}, *Age groups* = {0 - 4, 5 - 9, 90 - 94, 95+} and *Smokers* = {*no, passive, yes*}, then the CSS model would become $11 \times 2 \times 20 \times 3 = 1320$ times larger in terms of number of compartments/state variables in the CSS model. In such a situation, it becomes tempting to drop identity and use average values of, say, sexes and age

groups. However, CM-consistency may then not be achieved, and comparisons with ABM or EBM results will show differences because of over-simplification of the CSS model. The loss of identity also means that e.g. maximal or minimal through-times can usually not be measured because the individual token cannot be followed.

Furthermore, before the expansion, attributes that can take a continuous spectrum of attribute values must be discretised into a finite number of values, with an approximation accuracy that is sufficient for the purpose of the study. Space, therefore, is a very costly attribute in CSS modelling. The book *Modelling Biological Populations in Space and Time* [24] includes a good presentation on describing space in compartment-based models.

In [12], the best choice of model type (ABM, EBM or CBM/CSS) in different cases is discussed.

5. Stage-to-Compartment Expansion

5.1. Stage, Sojourn Time and Sojourn-Time Distribution

To understand and describe a system under study, we need a number of concepts. In the study of dynamic systems, *stage* is a crucial concept. After specifying what defines a stage, we can describe how continuous matter or discrete entities arrive in or leave the stage. Often the actual amount of content in a stage has a profound impact on how the system/model develops over time, but this amount depends on how long the matter or entities stay (*sojourn*) in the stage. Therefore, the *sojourn time* in a stage is often a crucial factor behind the dynamic development of a system under study that must be depicted in the model. For example, the duration of a pregnancy, the time as a pupa in an insect life cycle, the lifetime of a light bulb, the survival time in a disease or the time of a radioactive atom before decay can be important to describe in a dynamic model.

Furthermore, not all pregnancies, light bulbs and radioactive atoms have the same sojourn time. Therefore, average or expected values of the sojourn time in a stage are often used. However, averages are not always sufficient and it is often necessary to include the distributions of the sojourn time in a model to achieve proper results. Thus the *sojourn-time distribution or distributions*[3] is an important characteristic of a stage. The sojourn-time distributions describe how long a stage holds the matter or tokens (in deterministic or stochastic terms).

An example of the importance of a realistic description of the distribution of sojourn times is a SIR model with the stages: susceptible, infectious and recovered. Here the sojourn-time distribution of the infectious stage has a significant impact on the risk of an epidemic and its expected size.

Modelling the sojourn-time distribution may be necessary for some stages, but not for others. For example, the susceptible stage S of a SIR model is just a

---

[3]If there are several possible exits from a stage (e.g. progression, regression, death due to other cause etc.), then several sojourn-time distributions are defined for the stage, see [12].

source of individuals that may be infected, and the recovered stage R is just a container to count the cases that have had the disease. There is no need to describe the sojourn times for these stages. However, for the infectious stage I, the sojourn-time distribution is critical for the behaviour of the model, see Example A4.

It is also important to understand that a sojourn time is virtually never a physical constant and its distribution is rarely invariable. For example, in a study of productivity, the citizen in a society can e.g. be classified according to the stages: young, productive and retired. Then the productivity depends on the sojourn time in the productive stage. However, the sojourn time and its distribution in the productive stage depend on the length of education and the retirement age in the society at a particular time. It also depends on health, motivation, tax policy, competition, family policy and many other factors.

Even for a simple physical system under study, the sojourn time is not always a universal constant. For example, uranium-235 has a "natural" half-life of 704 million years, but in a reactor with plenty of neutrons the half-life is much shorter. In general, the sojourn time usually varies over time because of changes in the environment, and it may also vary because of e.g. cooperation or competition between studied objects. For example in a logistic model ($\Delta x/\Delta t = a \cdot x - b \cdot x^2$), the expected sojourn time varies with the nonlinear competition (the x^2-term). For a queuing system, the sojourn time in a queue will vary with a number of factors such as the arrival of customers, the number of service stations and the service time, which in turn may vary depending on the time of day and day of the week. By including these factors, we can still produce a realistic model.

Although the complexity of nature may seem overwhelming, an actual study is usually limited to describing the system under study under a set of specific conditions in order to fulfil a specific purpose. In short, the model should describe the structure, relations and quantitative values observed from the system under study in order to be able to reproduce the behaviour of this system. The variability of the world is not a problem for modelling a specific situation, but it means that generalization of the results of a model study is a complex task.

Regardless of whether the model is linear or non-linear, or affected by a varying environment or not, it is important that the model preserves the sojourn time and its distribution in the system under study for a well-defined situation or time period. This is equally true for all types of model studies, and CSS modelling is no exception.

5.2. Stage versus Compartment

Proper representation of the sojourn time and its distribution in a stage is an important task in all kinds of modelling. However, unlike a conceptual model or a DES model where a stage can be a building block with a specified sojourn-time distribution, CSS lacks the stage concept. Here a stage is *not* a component, so we

have to reconstruct or approximate a stage by a structure of compartments and transitions.

In CSS modelling, the *compartment* (*state variable*) is a box in which to place all matter or tokens with exactly the same set of attribute values, but the compartment will, in general, not produce the prescribed sojourn-time distribution of a stage. Therefore, the prescribed distribution must be generated by the building blocks available for a CSS model.

In order to limit the stage-to-compartment discussion, we now focus on some simple cases where the sojourn-time distribution is only a property of the matter or token. Some possible sojourn-time distributions are shown in **Figure 2**.

The single compartment, on the other hand, is not a flexible device to generate different types of sojourn-time distributions. In its simplest form, where the sojourn-time distribution is only a property of the matter or tokens $(\Delta x/\Delta t = -x/D)$, we obtain an exponential sojourn-time distribution, see **Figure 3**. For the discrete and stochastic case $(\Delta x/\Delta t = -Poisson[\Delta t \cdot x/D]/\Delta t)$ we expect the corresponding sojourn-time distribution.

Thus the use of a single compartment is generally insufficient for generating a prescribed sojourn-time distribution.

However, the compartment (state variable) is the fundamental "building block" of a CSS model. A compartment holds matter or tokens with exactly the
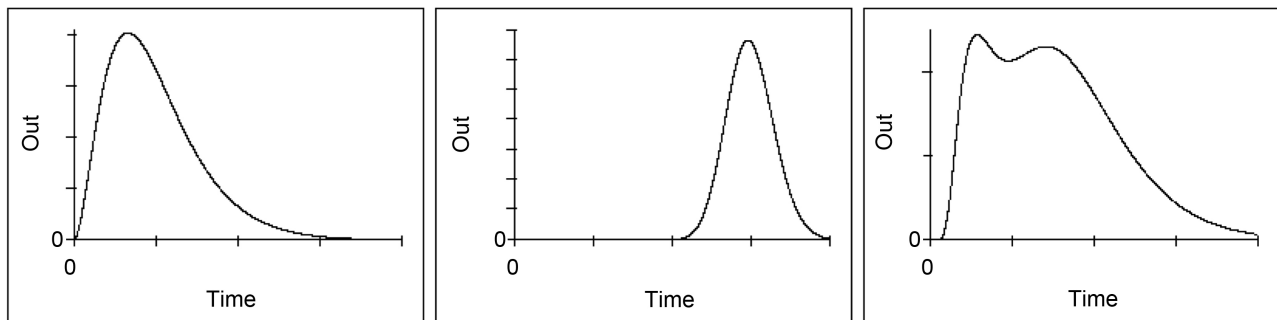
Figure 2. Some sojourn-time distributions of a stage. In the discrete and stochastic case these diagrams show the expected distributions.
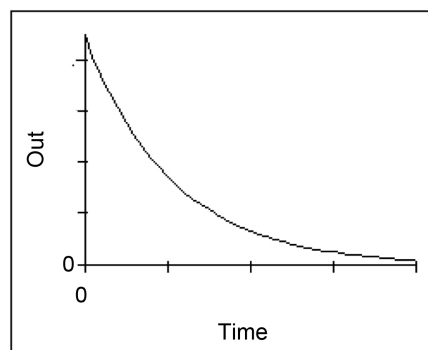
Figure 3. Sojourn-time distribution of a compartment where $\Delta x/\Delta t = -x/D$. In the discrete and stochastic case, this diagram shows the expected distribution.

same set of attribute values, except for identity and arrival time. The underlying idea of sacrificing identity is to be able to treat all matter or tokens in the compartment exactly the same and independently of how long they already have resided in the compartment. However, the sojourn after arrival must at least be approximately preserved by including an additional (local to the stage) sojourn attribute. In a simple case this can be approximately obtained by, say, three consecutive compartments with the local attribute values: newly-arrived, stayed-a-while, and soon-to-leave.

There are different ways to model a stage with a prescribed sojourn-time distribution by multiple compartments. For example:

- A shift register where the content is shifted one step for each time-step could be used. (Technically, this can be implemented as a circular buffer, and it is then enough to just shift the points of entry and exit.) This implementation has the disadvantage that the sojourn time will change when the size of the time-step is adjusted. (However, it would be possible to modify the implementation so that the sojourn time is independent on the time-step used.) See [8] for the stochastic case.

- A sub-model of compartments in series and/or parallel that generates or approximately generates the sojourn-time distribution of the stage can be constructed. This usually seems to be the best solution since it is general, in line with the CSS philosophy and can be directly applied in a CSS language. In this approach the number of compartments becomes independent of the size of a (sufficiently small) time-step.

Below we use the second approach, although we believe that implementation of a time-step independent shift register would be a convenient complement in some situations.

5.3. Realization of a Sojourn-Time Distribution in a CSS Model

In CSS modelling, the sojourn-time distribution of a stage can be generated from an adequate linear structure of compartments in series or parallel (or even using feedback). It is always possible to approximately generate any given sojourn-time distribution within a specified accuracy in this way.

Note that the same sojourn-time distribution of a stage is generated by a structure of compartments in both the continuous and the discrete cases. In the continuous case the structure generates the *fraction* of the content that will leave the stage at each time-step. In the discrete and stochastic case, the structure generates the *probability* of tokens leaving the stage at each time-step.

5.3.1. The Erlang Distributions (Compartments in a Series)

A common and often useful description of a stage is obtained by arranging k compartments (with the same (expected) exponential sojourn time D/k) in a series. This constitutes a "dynamic delay of order k" and is used so frequently in modelling that many CSS languages provide it as a building block where only the shape parameter k and the scale parameter (average/expected sojourn time) D

have to be specified.

Thus the sojourn-time distribution of a stage obtained from k serial compartments, each with an exponential sojourn-time distribution Exp(D/k), is denoted k-Erlang(D/k) distribution[4]. In **Figure 4**, the k-Erlang(D/k) distribution is shown for different values of k. Specifically, the *1*-Erlang(D) is the exp(D) distribution.

A serious but common error in CSS modelling is to represent a stage by a single compartment. A medical example of the disastrous consequences of confusing stage and compartment is given in Example A3, which describes the probability of a lesion developing into cancer. Using a single compartment to represent the lesion stage (called cancer in situ) in that case led to the wrong conclusion that cancer in situ of the cervix uteri is not a pre-stage of cancer that has to be treated. When this conclusion was applied in clinical practice, it led to the death of a number of women.

Another example is the SIR model of Example 1, above. Whether the infectious stage of this model (with a given average sojourn time D) is represented by a single or several compartments will drastically affect the expected size of an epidemic (see Example A4).

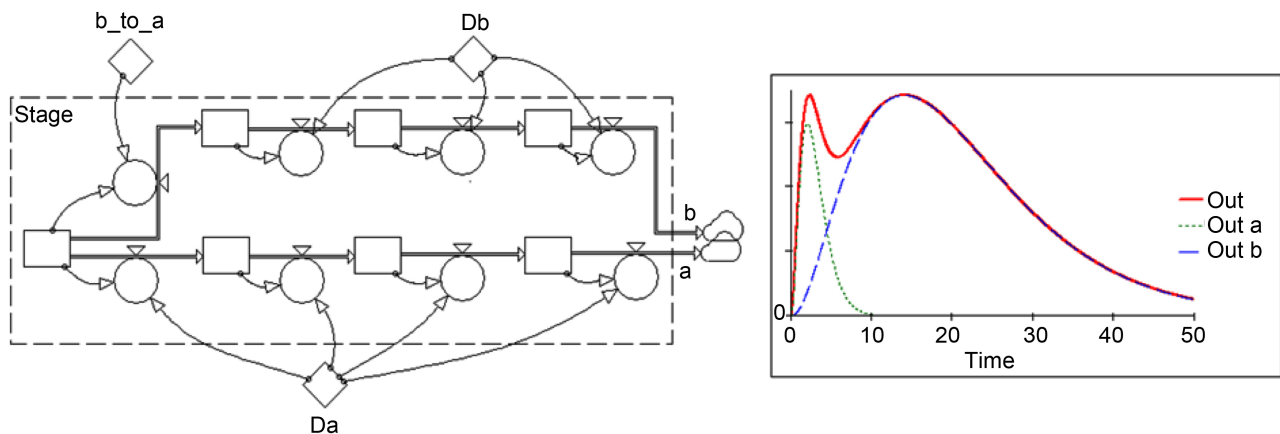### 5.3.2. Compartments in Series and Parallel

With compartments in series and parallel, any sojourn-time distribution can be approximated. **Figure 5** gives one example.

Roughly speaking, more compartments in series will concentrate the distribution around the sojourn-time average, while compartments in parallel will create
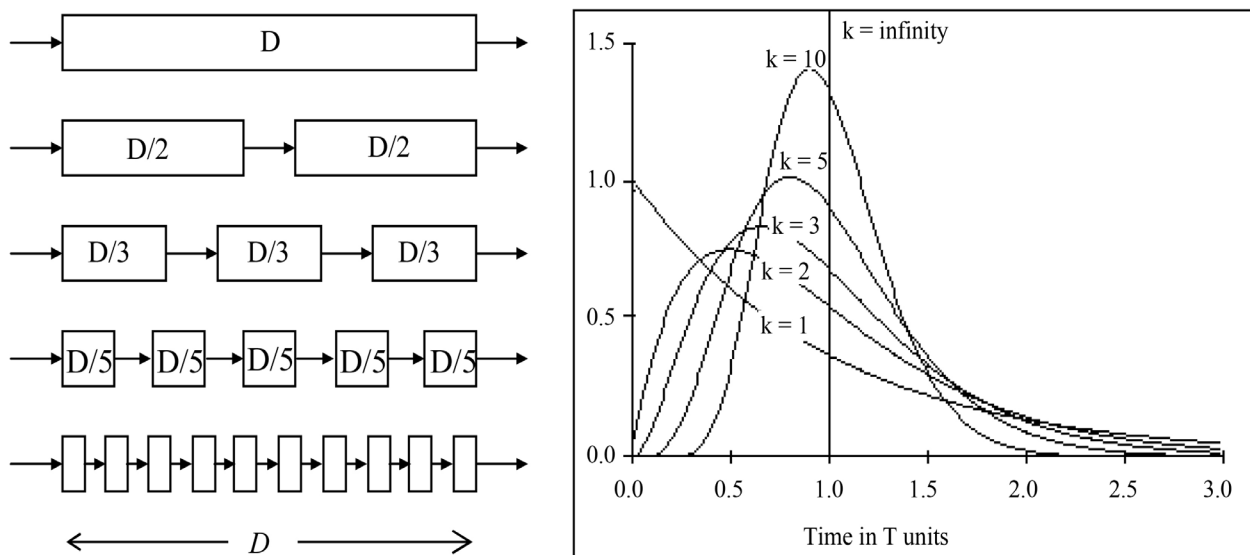
Figure 4. Left: A stage with an average sojourn time of D time units can be modelled by one or several compartments in series producing different sojourn-time distributions of the Erlang family. Right: The k-Erlang (D/k) family with an average sojourn time D and a shape parameter k for k = 1, 2, 3, 5, 10, and *infinity*. The sojourn-time distribution approaches a fixed time delay of duration D time units when k approaches infinity.

[4]For a positive integer k, the $\Gamma(k, \beta)$ distribution is called the k-Erlang (β) distribution, see [25] or a statistics textbook.

Figure 5. With structures of compartments in series and parallel, more complicated sojourn-time distributions can be created. (Here the input is divided into two parallel flows that produce the output. The parameter values $Da = 1$, $Db = 7$ and $b\_to\_a = 8$ are used in the plot.)

a hyper-exponential distribution.

5.3.3. The Impulse Response

A practical way to generate and to measure the sojourn-time distribution that works in the linear time-invariant case is to send an impulse into an empty stage (modelled by a sub-system of compartments). Then the outcome, called the *impulse response*, describes the sojourn-time distribution. However, in a non-linear case the outcome is dependent on e.g. the size of the impulse used, and in a time-variable case the outcome is what the environment imposes. For such cases, one can only talk about the actual sojourn time and its distribution in a specific situation.

6. Combined Simulation within CSS

The conventional way to include both discrete and continuous matter in the same model is to use combined DES and CSS. The theory behind this is thoroughly discussed by Zeigler *et al.* in their book *Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems* [26].

However, the concepts presented in this paper can be used to correctly handle both continuous and discrete parts in the same CSS model. Thereby, several advantages are obtained. First, a theoretically sound foundation is achieved. Second, combined problems can, in principle, be coded and studied exclusively in a CSS language, which is often considerably simpler than including DES. Third, the computational complexity is reduced when a large number of discrete objects can be modelled as transitions into and out of a relatively small number of compartments/state variables. Fourth, it also constitutes a flexible and execution-efficient way to model problems that would otherwise require a combined DES/CSS approach to preserve both the discreteness and the stochastics. Fifth, only one time-handling method (time-slicing) is involved, which is simple, safe

and fast.

In Example A6, an ecological model of a prey-predator system demonstrates how discrete predators feed on continuously varying biomass. Here, the discrete and continuous parts interact within the CSS concept.

However, there are special cases where the conventional CSS/DES approach is superior. In particular, the CSS/DES setting may be preferred to avoid large attribute expansions or complicated stage-to-compartment expansions. Moreover, when spatial attributes (x, y and z) are central properties of the studied objects, it is simpler and more exact to use DES for this than to expand the position attributes into a two- or three-dimensional grid of sub-models in CSS.

Another reason to use the combined CSS/DES approach is when the identity of the studied objects must be preserved, e.g. because the aim is to calculate minimal or maximal through times of the subjects in a queuing or a production system. Preserving the identity of the objects of a large population by attribute expansion would make the model huge and inefficient.

7. Handling Uncertainties

A model is always a simplified and incomplete description of a system under study because only the main components and relations that are important for the purpose of the study are included. Full and exact information about the structure and components of the system under study and about the system's behaviour is never available. We call this lack of information *uncertainty*.

With a *deterministic model*, we can describe the causal relations between different components of the system under study as we understand it. However, there is always irregular behaviour of the system under study that will not be reflected by such a model. Leaving out such irregularities may falsely remove a number of phenomena and may bias the results. This problem is partly described in Section 2.3, where modelling of discrete objects by a continuous model eliminates transition irregularities.

The irregularities of the system's behaviour contain information, although they may be far too complex to understand and model in detail. However, these irregularities can be at least partly reproduced by including random number generators for appropriate statistical distributions in the model. The resulting *stochastic model* will then behave more similarly to the system under study and the statistical estimates from the model will be more relevant.

7.1. Different Types of Uncertainty

How uncertainties are modelled can strongly affect the results and conclusions of a model study, and it will certainly affect the confidence intervals for the results.

We start by discussing where uncertainty can appear in a CSS model. Then in the following subsections, we discuss how these types of uncertainty can be handled.

Of course, there can be uncertainty about the overall structure of the model, so-called *structural uncertainty*. We can also focus on the components of a CSS model that (graphically) are composed of compartments, physical transitions, information transitions and parameters. Each of these types of components can be related to an uncertainty.

Figure 6 demonstrates the locations of the uncertainties in a CSS model, graphically represented by a *Forrester diagram* [27]. Here an oversimplified SIR model, similar to that in **Figure 1**, is used to make the example more concrete. To include all types of uncertainty into this example, we add an authority that collects information about the number of infectious individuals and issues recommendations about avoiding contact and using sanitation measures in order to reduce the infection rate.

The epidemic system under study in **Figure 6(a)** is represented by the dynamic model inside the dashed rectangle in **Figure 6(b)**. In this *core part* of the model, dynamic relations of interest are described. However, every transition of the infection because of who meets whom, how they then behave etc. will not be known. We then have to describe the uncertainties about the transitions in stochastic terms. Similar uncertainties are associated with the information collected, analysed and issued to the population.

A system under study is almost never self-contained. The spread of the epidemic and the time to recover are affected by people's behaviour, which in turn depends on the weather, season of the year, recommendations etc. All such factors,
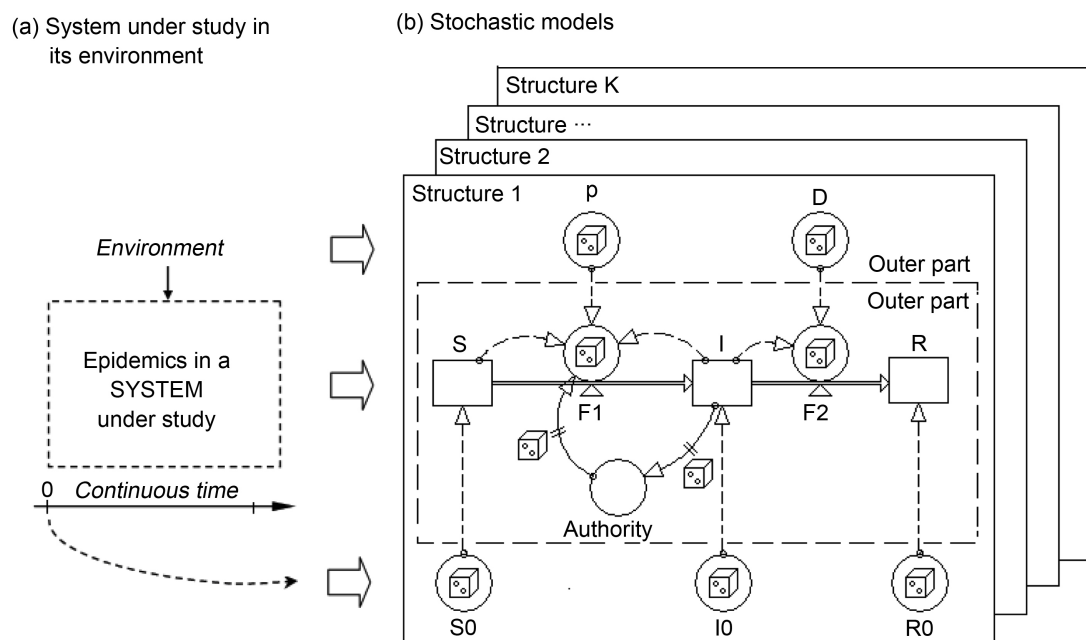
Figure 6. (a) An infectious system under study; (b) Alternative epidemic models, the first of which is a SIR model based on the stages *S*, *I* and *R* (initiated to *S*0, *I*0 and *R*0), transitions *F*1 and *F*2, and parameters *p* and *D*. An "Authority" collects information about the number of infectious individuals and issues recommendations, which affect the infection rate. Dice show where different types of stochasticities can be included. (The different types of information links are described in Section 7.5.)

too complicated to understand and model in sufficient detail, lie outside the system boundaries and thus belong to the *environment* of the system under study. The environmental impact on the system can, however, be measured and described in statistical terms.

These unexplained influences of the environment on the system under study are introduced in the model as parameters. Furthermore, the conditions at time zero, depending on an unexplained prehistory, are described in terms of initial values, if necessary in stochastic terms. Therefore, we locate the parameters and initial values in the *outer part* of the model in **Figure 6(b)**.

Now we comment on the five types of uncertainty of the model:

1) *Structural uncertainty* about the system under study can be handled by alternative models. The uncertainty is then reflected by the difference in behaviour of these models.

A CSS model is composed of compartments, transitions, information links and parameters. Each of these constituent elements can have its own type of uncertainty:

2) Approximate information about the number of objects in the stages at time zero can motivate *initial value stochasticities* for $S(0)$, $I(0)$ and $R(0)$.

3) There is almost never enough information to decide the exact event times for transitions of tokens. Therefore, *transition stochasticity*[5] in the flows $F1$ and $F2$ is used to complement the dynamic description with available statistical information about the transitions.

4) Uncertainty about the delay of information can be described by *information stochasticity*.

5) Unexplained irregularly varying values of the parameters p and D motivate the use of *parameter stochasticity*[5] to generate realistic inputs from these parameters. (If it is uncertain e.g. how the information will affect the Authority's recommendations, we could also include a stochastic parameter linked to the Authority.)

The locations of the stochasticities are symbolised by dice in **Figure 6(b)**. ∎

We now proceed with a discussion of the five types of uncertainty and how they can be handled in CSS modelling.

7.2. Structural Uncertainty

"Structure 1" of the SIR model in **Figure 6(b)** is just one possibility. In particular, the sojourn time in the infectious stage is here modelled by a single compartment, implicitly assuming its sojourn-time distribution to be exponential. Other assumptions about this distribution can be realised by using several compartments in parallel and/or series, as discussed in Section 5.

Another structural possibility is that an infected individual does not immediately become infectious. The exposed individual may require a latent period

---

[5]*Transition stochasticity* is often called *demographic stochasticity* and *parameter stochasticity* is often called *environmental stochasticity* in biological and ecological population models, see e.g. [28] [29].

before entering the infectious stage. This can be accomplished by including an Exposed (E) stage between the Susceptible and the Infectious stages (giving a so-called SEIR model [30]).

By studying the outcomes (size of the epidemic, time until the epidemic is over etc.) for the alternative models (after parameter estimation), it is possible to choose the model structure that best agrees with observations of the real system's behaviour.

7.3. Initial Value Uncertainty

Uncertainty about initial values of the compartments can be handled by generating variability of the initial conditions *in repeated simulation runs* (replications). *Initial value stochasticity* operates before the start of the execution and does not intervene during the replication.

Consideration of initial value stochasticity may be important when the initial situation of a studied process is unclear because of inaccurate information. For example: How many in a studied population were initially susceptible, infectious and recovered (*i.e.* immune)? Or will the next epidemic start from a single infectious individual or from a batch of infectious individuals returning from abroad?

7.4. Transition Uncertainty (Exists only in the Discrete Case)

The transitions of tokens in a discrete model may be regularly scheduled over time or may take place when some condition is satisfied. However, often the transitions must be regarded as irregular, *i.e.* treated as random with a specified intensity. In this section, we discuss only the latter case.

With time-slicing the number of transferred tokens is updated for each (sufficiently small) time-step Δt. In the general case, we have a time-varying Poisson process implying that the number of transiting tokens during Δt is Poisson distributed according to: $Po\big[\Delta t \cdot \lambda(t)\big]$, where the intensity $\lambda(t)$ can be any function of time, including e.g. stochastic parameters and state variables. The rationale behind this is treated in Section 2.2, while Examples A1 to A6 illustrate behaviours of models with transition stochasticity.

Technically, the number of tokens transferred $\big(Po\big[\Delta t \cdot \lambda(t)\big]\big)$ during a short time-step Δt is obtained by drawing a random number from a Poisson distribution with $\Delta t \cdot \lambda(t)$ as argument. (When a stage is expanded into a structure of compartments, there will also be internal stochastic transitions between the compartments representing the stage.)

Transition stochasticity has a nice property inherited from the Poisson distribution: $Po\big[\Delta t \cdot \lambda\big] = Po\left[\frac{1}{2}\Delta t \cdot \lambda\right] + Po\left[\frac{1}{2}\Delta t \cdot \lambda\right]$, which means that the *expected* outcome is independent of the size of the time-step used.

Note that the transition stochasticity generates both the dynamics and the stochastic development of a discrete model *in an inseparable way*. This inseparability is important. There are many awkward examples of trying to separate

dynamics and stochastics by e.g. adding stochastic noise to a deterministic model, which creates a number of artifacts (see Example A1).

As discussed in Section 2, the Po[ ] clause will vanish in the embedded continuous model, leaving only the argument $\Delta t \cdot \lambda(t)$, which has no transition stochasticity. However, a continuous flow can still be stochastic because of unexplained variations in the parameters, e.g. $\lambda(t) = p(t) \cdot X(t)$, where $p(t)$ is a stochastic parameter and X is the content of a compartment/state variable. This is treated in Section 7.6.

7.5. Information Uncertainty

Information uncertainty is similar to transition uncertainty, but here continuous or discrete information-bearing signals, instead of physical matter, are transferred. However, before discussing this we must clarify that "information" is an ambiguous concept in CSS modelling.

In addition to "*real* information", where an information bearer transfers information, there is also "*artificial information*" that connects artificially separated building blocks of the model.

For example, when a radioactive atom decays, there is just one radioactive atom less in the system under study. However, in a CSS model this is described by both a compartment and a physical outflow (and more generally in CSS, by both a state equation and a flow equation). An artificial information link from the compartment to the valve regulating the outflow then has to be included. Its only function is to transfer logical information from the compartment equation to the flow equation. The "artificial information" is immediate, intrinsic, cannot be delayed, distorted or associated with uncertainty, and it has no counterpart in the system under study. In Figure 6(b), we intentionally denoted the artificial information links as dashed links (- - ->). Such links are the connections from parameters p and D to the flows $F1$ and $F2$, the links from the initial values $S0$, $I0$ and $R0$ to the compartments, the links from the compartments S and I to the flow $F1$, and the link from compartment I to flow $F2$.

Real information, on the other hand, is information that is *physically* communicated in the system under study. It may contain information about objects or situations or about changing external conditions. The information can affect other parts of the system (e.g. a stop signal, or a signal as part of a control system). In Figure 6(b), we modelled real information by solid links (—>). Such information has an *information bearer* that can be continuous or discrete.

In Figure 6(b), an "Authority" collects information on the number of infectious individuals and issues recommendations in order to reduce the infection rate. Here information from the Infectious compartment to "Authority" and from "Authority" to the transition flow, $F1$, describes real information flows *in the system under study*. The transfer of this information requires time that may vary in an unpredictable way. (If the information is sent by post, it may not be delivered during the weekend, while recommendations sent to the media are

consumed at different times by the individuals.) The information may also be distorted for various reasons (imperfect information, typing error, distortion of the message, misinterpretation, not reaching all subjects).

Thus information bearers can be *delayed* and the message may be *distorted*. An information delay can be modelled in CSS in a very similar way to a material delay, where a structure of compartments and transition flows creates a so-journ-time distribution, but here an information bearer is delayed. If uncertainty in the delay time is involved here, we can include *information stochasticity* in exactly the same way as for transition stochasticity. Information may also be deliberately or randomly distorted. For example, the "Authority" in Figure 6(b) may distort the information in a way we cannot model but describe statistically. Then we can include a parameter that uses this statistical information and affects the "Authority".

7.6. Parameter Uncertainty

Parameter uncertainty is discussed in Section 4.3.5 of [12]. The relevant part of that discussion is reproduced below.

Uncertainty about parameter values can be handled by *parameter stochasticity*. Unlike transition stochasticity, which is intrinsic in the time-handling method and is statistically completely specified to its form, parameter stochasticity is used to describe the lack of knowledge about how the environment affects the system under study. This lack of knowledge may be about unknown constants or about unexplained variations that we perceive as irregularities. Such irregularities may be variations in temperature, wind, precipitation, food supply, pollution, fertility, noise or disturbances of any kind. To include such external irregularities, these either have to be generated by an adequate sub-model (which is often not possible) or the unexplained irregularities have to be described by appropriate, often empirical, statistical distributions.

Parameter stochasticities are introduced in the parameters by making them stochastic. This is shown in the SIR model example in Figure 6(b) by placing the risk parameter, p, and the sojourn time parameter, D, in the outer part of the model as parameter inputs to the model. Thus, for the SIR model, parameter stochasticity will affect the development *indirectly*, via parameters that in turn influence the transition probabilities.

Technically, the values of a parameter describing e.g. morbidity, mortality, fertility, risk, or sojourn time have to be drawn from an appropriate statistical distribution during the replication. When a parameter is constant but unknown, it resembles the initial value case, requiring only one random sample from a statistical distribution before the onset of a replication. However, when the parameter changes irregularly over time (e.g. temperature or precipitation) there are two stochastic aspects involved: "*When to change the parameter?*" and "*What is the new value?*". Samples from two random number generators, in general with different distributions, are then required.

This illustrates a specific problem involved in parameter stochasticity. Its effect depends on the time-step used if you don't control it. Therefore, you have to regulate "*when to change the parameter values*". Otherwise a new random sample is drawn for the parameter at each time-step.

7.7. Correlations in Parameters and Initial Values

Various measured quantities are often *correlated* because of the way objects interact with each other. Correlation between A and B can emerge for different reasons: causally because A may affect B, B may affect A, or A and B may affect each other. However, there may also be a C affecting both A and B without any causal connection between A and B. Finally, an observed correlation may occur by chance.

Correlation between $A(t)$ and $A(t - T)$ is called *auto-correlation*, while correlation between $A(t)$ and $B(t)$ is called *cross-correlation*. For example, the weather on one day may be strongly auto-correlated to that on the previous day, and the amount of sunshine and the amount of precipitation may have a negative cross-correlation during daytime. In principle, both types of correlation can be positive or negative.

In statistics, correlations often play an important role in describing a system under study.

In modelling, on the other hand, we strive to describe the processes in the form of causally interacting components, which *generate* correlations in observed model quantities. A model where the structure of the system under study is preserved will recreate the proper dynamics, including correlations. However, this is only the case for the core part of the model describing the processes within the system borders. The model must usually also include a description of how environmental quantities affect the core part of the model. (Note here that the system borders not only define what is included in space and time, but also define the level of detail described in the model.)

In the SIR model shown in **Figure 6(b)**, the parameters p and D could be cross-correlated e.g. because of the weather (e.g. if the weather is bad, people may meet indoors where the risk p is higher and the sojourn time D in the infectious stage may be longer). Further, the initial values of the sub-populations $S0$, $I0$ and $R0$ must add up to the total population N. If there is uncertainty about the initial number of susceptible individuals, we can use initial value stochasticity to initiate the SIR model with different values of $S0$ for different replications, but then $S0$ becomes negatively cross-correlated to $I0$ and $R0$ so that the total population is preserved.

Correlation between stochastic parameters or initial values can be created in different ways:

1) Using statistical time series from real observations.

2) Drawing random numbers from a joint (multivariate) distribution to obtain cross-correlation.

3) Weighting together a new random sample with previous random samples, to obtain auto-correlated parameter values.

4) Modelling the process that generates the correlation (which means also changing the system boundaries).

7.8. Random Number Generators

To perform stochastic modelling and simulation, high quality random number generators (RNGs) for various distributions must be available in the CSS language used.

The RNGs should have good statistical qualities, a long period and be fast. Furthermore, it is valuable if a seed can be set to obtain reproducibility of the replication. A seed also enables the use of variance reduction based on e.g. *Common Random Numbers*. Other variance reduction techniques, e.g. *Antithetic sampling*, can sometimes also be a good asset if supported by the RNGs (see [16] [25] [31]).

Most CSS languages contain a set of RNGs such as Uniform, Poisson, Binomial, Normal, Exponential etc. In some languages an RNG for empirical distributions that the user specifies is also provided. This makes it technically simple to implement the different types of stochasticity described above in a CSS model.

8. Statistical Output Analysis

The very essence of a stochastic model is that the outcome of a replication is unpredictable, so the results differ between replications. Therefore, a large number of replications must be performed and the results from these collated, statistically analysed and presented. Another way to put it is that the result from a deterministic model is a number, while the corresponding result from a stochastic model is a distribution. So several results from a deterministic model form a vector of numbers, while the corresponding results from a stochastic model make up a joint distribution.

A consequence of this is that even when the time to execute a stochastic model is only slightly longer than that for executing the corresponding (embedded) deterministic model, the requirement of 100 to 10,000 replications of the stochastic model to obtain statistical estimates makes a substantial difference.

Before discussing how a stochastic outcome can be analysed and presented in statistical terms, we must distinguish two kinds of statistics from a simulation study based on a stochastic model.

The first kind of statistics we denote *internal*, because it concerns the stochastic variations *within* a replication. Then data must be collected during the replication. These data are used to calculate various statistical estimates (average, standard variations, min, max etc.). It is often convenient to make these statistical calculations *during* the replication by counting events, summing up quantities X and their squares X^2, testing if X is the largest or smallest value so far etc.

The second kind of statistics we denote *external*, because it concerns the sto-

chastic variations *over many replications*. Then data collection from many replications, followed by a statistical analysis and presentation, has to be done.

Observe that there is one principal difference in the calculation of a statistic measure within a replication and over N replications. Within a replication, the conditions change dynamically so the samples are *not independent* and not drawn from an identical distribution, whereas a number of repeated replications (of the same model under the same experimental conditions) are *independent experiments*. In statistical terms, we obtain results from *independent and identically distributed* (*iid*) stochastic variables in the external case. Except for cases when this is important, the statistical formulae remain the same.

These types of statistics can be nested; for example, one can calculate the maximum *within* each of N replications and then calculate the average of maxima *over* these N replications.

In CSS modelling and simulation, a CSS language can be used. However, this type of language is generally used for deterministic modelling and simulation, so statistical devices are usually lacking. Therefore, methods and devices, both for internal statistics within and external statistics over many replications, are required.

In this section, we first focus on the types of statistical estimates that can be of interest in a stochastic simulation study. We then discuss how to obtain the statistics *within a replication* and *over many replications* in stochastic CSS modelling.

8.1. Statistical Estimates

In a stochastic modelling study, one is typically interested in estimates of different kinds of quantities. Such quantities can be amount of matter or number of tokens in a stage, or number of events of a certain kind that can be accumulated by counters. It can also be the min, max or average of a quantity. One may also be interested in estimating a risk or the average sojourn time of a stage.

Queuing for resources may also be studied with a stochastic CSS model. Then queuing estimates such as number of tokens waiting for a resource, their waiting times, the utilization of resources such as taxi cabs, doctors etc. may be of interest, as well as economic aspects.

8.1.1. Probability Distribution/Density Function (p.d.f.)

Assume that we have recorded N observations of m different quantities $X_1, X_2, \cdots, X_m$. The *total information* collected can then be expressed in the form of a *joint probability function* (joint p.d.f.).

The p.d.f. of a single quantity X_i is the marginal distribution of the joint distribution. This p.d.f. can be presented as a histogram or used to calculate average, standard deviation, confidence interval, minimum, maximum etc. of X_i.

A joint distribution can also be used to study e.g. *covariance* or *coefficient of correlation* between quantities.

Depending on the purpose of the study, there can be many different types of

performance measures to be estimated. These estimates can be expressed as *point estimates* or *interval estimates*. *Testing of hypotheses* can also be done.

8.1.2. Point Estimates

Point estimates are obtained from sampled data to calculate a single value of a studied quantity. Such estimates are *average value, median value, standard deviation, final value at the end of the simulation, correlation, min* or *max value* etc.

A direct way to calculate many point estimates of quantities X and Y is based on simple formulae that are updated for each new value. For example, one can sample values for the quantities X and Y and accumulate ΣX, ΣY, ΣX^2, ΣY^2 and $\Sigma X \cdot Y$. This is sufficient for calculating averages, variances (and standard deviations), correlation between X and Y etc. Minimum and maximum values are also easily obtained by checking if the current X is the smallest/largest value so far.

8.1.3. Interval Estimates

Interval estimates are obtained from recorded data of a quantity to calculate an *interval* of probable values, in contrast to a point estimate, which is a single number. Examples of interval estimates are *confidence interval* and *tolerance interval*.

An interval I_X (written (I_l, I_u) for lower and upper limits) that with probability $1 - \alpha$ (e.g. 95%) will cover the outcome of a quantity X is a *confidence interval* (C.I.) of X at the confidence level $1 - \alpha$. If the interval is a *1-sided* interval, e.g. $(0, I_u)$, the probability of X being above the interval is α. For a 2-*sided* interval (I_l, I_u), the interval is usually located so that the probability of X being below I_l is $\alpha/2$ and of being above I_u is also $\alpha/2$.

It is important to carefully distinguish between two cases of confidence intervals: It may be used for an estimate of how much X varies, or it may be used to estimate how precise the *average of X* is.

In the first case, a confidence interval of X can easily be obtained from the p.d.f. from e.g. 10,000 observations. At e.g. 95% confidence level, we search for the lower and the upper limits given by the 250th and the 9750th observations of X (sorted by size), respectively. In this case, the number of observations will *not* affect the (expected) size of the confidence interval.

In the second case we want a confidence interval around the estimated *average of X*, $X_{av} = \sum X_i / N$, obtained from, say, N = 10,000 observations. When the N observations are independent and identically distributed (*iid*), then the Central Limit Theorem applies, and *the sum* of N *iid* random variables will approach a *Normal* (X_{av}, σ) distribution as N increases (even when the distribution of X is very different from Normal). A 2-sided confidence interval around X_{av} at a confidence level of $1 - \alpha$ can then, for large N (say > 20), be approximated by: $Ix_{av} = \left(X_{av} - \lambda_{\alpha/2} \cdot d, X_{av} + \lambda_{\alpha/2} \cdot d \right)$; where $\lambda_{\alpha/2}$ = 1.96 for a confidence level of 95%, and $d = \sigma / \sqrt{N}$ (or $d = s / \sqrt{N}$, when σ is approximated by the obtained estimate s of the standard deviation).

Note that d decreases as $1 / \sqrt{N}$, which means that the confidence interval

around X_{av} can take any small width provided that the number of observations, N, is sufficiently large. In fact, this is true *for the model*, but not for the system under study. The model is a simplified description of the system under study and will always deviate from the real system in a number of ways. Simulating the model for more observations will produce more exact estimates of the *model's* behaviour, but this should *not* be confused with the behaviour of the *system under study*.

Another treacherous property of confidence intervals is that the fewer uncertainties that are recognised and included in the model, the smaller (sic!) the confidence intervals obtained, which once again may give a false impression about the precision of the estimates from the model.

8.1.4. Test of Hypotheses

A hypothesis can be tested in the following way: Make a *null-hypothesis H_0* that states something you hope to be able to reject. The *alternative hypothesis H_1* is the complementary statement to H_0. The null hypothesis will be *rejected* only if the observations suggest that H_0 is false with a probability of at least (say) 95%. (H_0 is then rejected at a *confidence level* $1 - \alpha = 0.95$.)

There is a close relationship between *testing of a hypothesis* and a *confidence interval*. A test of a hypothesis according to the "confidence interval" method proceeds in two steps:

1) First calculate a confidence interval on the desired confidence level $1 - \alpha$.

2) If the tested quantity is outside the interval, H_0 is rejected at confidence level $1 - \alpha$.

8.2. Internal Statistics

Estimates *within* a replication are based on sampling of quantities during the replication. Since the observations come from a dynamic and stochastic process, there is usually a dependence between $X(t)$ and $X(t + \Delta t)$ etc. as well as between $X(t)$ and another variable $Y(t)$. In other words, the *iid* property is *not* fulfilled.

Average, standard deviation, covariance and correlation coefficient can be calculated after the replication, but it is usually more convenient and efficient to update the estimates of averages, standard deviations, co-variances, minima and maxima etc. for each new sample. For this, X, Y, X^2, Y^2 and $X \cdot Y$ can be accumulated. **Figure 7** demonstrates how averages, variances, standard deviations, co-variance and correlation can be calculated during a replication. The key mechanism here is to update ΣX, ΣY, ΣX^2, ΣY^2 and $\Sigma X \cdot Y$ over time by using the standard compartment and transition elements of a CSS language.

The formulae behind **Figure 7** are straight-forward (where Z denotes X, X^2, Y, Y^2 or XY):

$$\text{CUM\_Z} := \text{CUM\_Z} + \text{DT} * Z \qquad \text{(with CUM\_Z initialised to zero) (3a)}$$

$$\text{E\_Z} := \text{CUM\_Z}/\text{Time} \qquad \text{(Time average)} \qquad (3b)$$
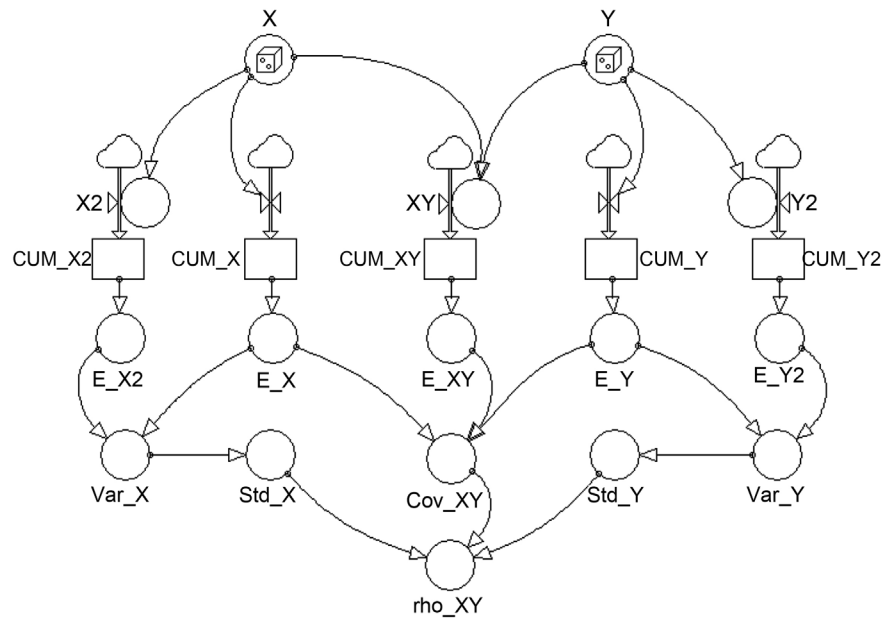
Figure 7. Calculation of averages, variances, standard deviations, covariance and coefficient of correlation within a replication. This can be accomplished by the building blocks in your CSS package.

$$\text{Var\_Z} := \text{E\_Z2} - \text{E\_Z} * \text{E\_Z} \qquad \text{(Variance)} \qquad (4a)$$

$$\text{Std\_Z} := \text{SQRT}(\text{Var\_Z}) \qquad \text{(Standard deviation)} \qquad (4b)$$

$$\text{Cov\_XY} := \text{E\_XY} - \text{E\_X} * \text{E\_Y} \qquad \text{(Covariance)} \qquad (5a)$$

$$\text{rho\_XY} := \text{Cov\_XY}/(\text{Std\_X} * \text{Std\_Y}) \qquad \text{(Coefficient of correlation)} \qquad (5b)$$

These calculations can be coded as a macro if the CSS language has this facility.

Devices for finding the smallest or largest value of a quantity *X* during a replication are often available as library functions in the CSS language. Otherwise, they can easily be constructed by checking for a new smallest or largest value of *X* at each time-step, in which case the new value becomes the candidate to be compared with future samples.

Example A5 shows how a queuing system can be modelled. It also demonstrates how to include devices for queuing measures, such as averages for queue length, waiting time and utilization of a resource.

8.3. External Statistics

CSS languages give no support for estimates over many replications, since they are mainly intended for deterministic modelling. Manually performing hundreds or thousands of replications, collecting the results from each replication, performing statistical analyses of the results and presenting the statistical results would be extremely tedious and time consuming.

CSS languages often provide the possibility to export results after execution. Then the results can be exported to e.g. a spreadsheet and the statistical facilities

of this spreadsheet can be used.

However, to work efficiently, a powerful tool to order a series of N replications, collect results from each replication, perform statistical analyses of the results and present the statistical results is required.

8.3.1. Philosophy behind a Tool for External Statistics

Before constructing a tool to produce external statistics from a series of replications of a CSS model, it is important to have a clear philosophy of what to handle by this "tool" and what to handle in the CSS model.

The tool should be a programme that controls the execution of the CSS model and performs the following functions:

- Specify and connect to a certain model file.
- Specify the variables to be analysed.
- Order a number of N replications of the model.
- If desired, make the study reproducible by controlling the sequence of seeds for the N replications.
- Retrieve the results of the specified variables for each replication.
- Make statistical analyses of the results from the N replications.
- Present the statistical results in appropriate ways.

No intervention by the tool should be made during the replications. The communication between the tool and the model should be restricted to before and after full replications. The mechanisms in the tool should be few and simple.

8.3.2. A Tool for External Statistics

A tool for handling, analyzing and presenting statistics from multiple simulations of a stochastic CSS model [32] was constructed for the CSS language Insight Maker [33]. Because Insight Maker is not fully open source, we constructed a rudimentary CSS language, **StochSD** (Stochastic System Dynamics), based on the open source part of Insight Maker where we included the statistical tool, called **StatRes**.

StatRes (Statistical Results) connects to the current model file, orders N replications, collects results of specified quantities and calculates statistics on averages, standard deviations, confidence intervals, min & max values and percentiles. It can also present the statistics in the form of histograms, box plots and scatter plots (X-Y plots) with coefficient of correlation between X and Y calculated, etc.

StatRes can also control the seeds used for the replications. This feature can be used for reproducibility and for variance reduction with e.g. Common Random Numbers.

For using **StochSD** (including **StatRes**) over the net or downloading it, go to: https://stochsd.sourceforge.io/homepage/. For more details see **Appendix B**: Supplementary material.

9. Discussion and Conclusions

Since the introduction of different kinds of numerical simulations in the 1950s

and 60s, the merits of CSS and DES have been discussed, sometimes in terms of micro and macro modelling and simulation. The underlying problem has been that micro models (DES) and macro models (CSS) often do not produce consistent results, leading to questions about which type of modelling is correct [2] [3] [4] [5] [6] [11] [34] [35].

Nowadays, we know the cons and pros of each approach [10] [11] [12]. We understand why conventional CSS modelling can distort results, and we know how to perform CSS modelling in a correct way. We also know how to translate between DES and CSS models without losing consistency, and we understand the consequences in complexity, size and execution time for the different choices, see [12]. This takes the mystery out of the micro/macro debate in a constructive way.

It is clear that the old CSS approach of modelling all quantities as continuous, often without proper attribute and stage-to-compartment expansions and without a good understanding of how to implement uncertainties, is very limited in its possibilities to produce CM-consistent models.

A central question which this paper discusses is: *Is it theoretically and practically possible to produce CM-consistent models within CSS modelling and if so, what is required*?

The answer is that CM-consistency is theoretically possible to attain, provided that the modeller knows how to model continuous matter and discrete tokens, how to make attribute expansions, how to perform stage-to-compartment expansions to obtain the intended sojourn-time distributions and how to implement different types of uncertainties. From a practical point of view, this is even relatively straightforward for many models. However, some models, with e.g. many attributes or attributes that can take many (or infinitely many) values, are best handled by DES modelling.

A skilful modeller should also know the prerequisites for using one type of modelling instead of another, as well as the consequences of the respective choice. The greatest asset of CSS is that it is almost independent of the size of material or population (provided that the identity of described objects can be dropped), while the weakest part is the cost of handling attributes and sojourn-time distributions. For the DES approach, on the other hand, the model size grows linearly with the number of objects described, but it can easily accommodate many attributes (even unique identities and space coordinates), and an agent's or entity's sojourn time in a stage is handled without any expansion by drawing a single random number from an appropriate distribution.

Furthermore, consistent CSS modelling with both continuous and discrete parts can in many cases replace the often awkward combined modelling that intermixes concepts from the CSS and DES approaches. Even though many CSS or DES languages contain construction elements from the alternative approach, the understanding and validity of a model built on a combination of CSS and DES concepts can be unsatisfactory or obscure.

This paper is intended as a guide to consistent CSS modelling and we believe that its content should be included in basic courses in CSS.

We have also provided some supplementary material. First, the models presented in Appendix A are accessible over the internet from where they can be downloaded or directly executed. Second, an open source software tool for statistical post-analysis of results from many replications is also provided, see Appendix B.

If the recommendations in this paper are followed, CSS modelling will reach its full potential and many unintended inconsistencies can be avoided.

Acknowledgements

References

[1] Kreutzer, W. (1986) System Simulation: Programming Styles and Languages. Addison-Wesley, Sydney.

[2] Fahse, L. Wissel, C. and Grimm, V. (1988) Reconciling Classical and Individual-Based Approaches in Theoretical Population Ecology: A Protocol for Extracting Population Parameters from Individual-Based Models. *The American Naturalist*, **152**, 838-852.

[3] Gómez-Mourelo, P. (2005) From Individual-Based Models to Partial Differential Equations. An Application to the Upstream Movement of Elvers. *Ecological Modelling*, **188**, 93-111. https://doi.org/10.1016/j.ecolmodel.2005.05.014

[4] Grimm, V. (1999) Ten Years of Individual-Based Modelling in Ecology: What Have We Learned and What Can We Learn in the Future? *Ecological Modelling*, **115**, 129-148. https://doi.org/10.1016/S0304-3800(98)00188-4

[5] Huet, S, and Deffuant, G. (2008) Differential Equation Models Derived from an Individual-Based Model Can Help to Understand Emergent Effects. *Journal of Artificial Societies and Social Simulation*, **11**, 10. http://jasss.soc.surrey.ac.uk/11/2/10.html

[6] Wilson, W.G. (1998) Resolving Discrepancies between Deterministic Population Models and Individual-Based Simulations. *The American Naturalist*, **151**, 116-134.

[7] Gustafsson, L. (2000) Poisson Simulation—A Method for Generating Stochastic Variations in Continuous System Simulation. *Simulation*, **74**, 264-274. https://doi.org/10.1177/003754970007400501

[8] Gustafsson, L. (2003) Poisson Simulation as an Extension of Continuous System Simulation for the Modeling of Queuing Systems. *Simulation*, **79**, 528-541. https://doi.org/10.1177/003759703040234

[9] Gustafsson, L. and Sternad M. (2007) Bringing Consistency to Simulation of Population Models—Poisson Simulation as a Bridge between Micro and Macro Simulation. *Mathematical Bioscience*, **209**, 361-385. https://doi.org/10.1016/j.mbs.2007.02.004

[10] Gustafsson, L. and Sternad, M. (2010) Consistent Micro, Macro and State-Based

Modelling, *Mathematical Biosciences*, **225**, 94-107.
https://doi.org/10.1016/j.mbs.2010.02.003

[11] Gustafsson, L. and Sternad, M. (2013) When Can a Deterministic Model of a Population System Reveal What Will Happen on Average? *Mathematical Biosciences*, **243**, 28-45. https://doi.org/10.1016/j.mbs.2013.01.006

[12] Gustafsson, L. and Sternad, M. (2016) A Guide to Population Modelling for Simulation, *Open Journal of Modelling and Simulation*, **4**, 55-92.
https://doi.org/10.4236/ojmsi.2016.42007

[13] Gillespie, D.T. (2001) Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems. *Journal of Chemical Physics*, **115**, 1716-1733.
https://doi.org/10.1063/1.1378322

[14] Devore, J.L. (2004) Probability and Statistics for Engineering and Sciences. 6th Edition, Thomson Learning, Inc., Toronto.

[15] Hahl, S.K. and Kremling, A. (2016) A Comparison of Deterministic and Stochastic Modeling Approaches for Biochemical Reaction Systems: On Fixed Points, Means, and Modes. *Frontiers in Genetics*, **7**, 157. https://doi.org/10.3389/fgene.2016.00157

[16] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (2007) Numerical Recipes: The Art of Scientific Computing. 3rd Edition, Cambridge University Press, New York.

[17] Braun, M. (1993) Differential Equations and Their Applications. Springer-Verlag, New York. https://doi.org/10.1007/978-1-4612-4360-1

[18] Chatterjee A., Vlachos D.G. and Katsoulakis, M.A. (2005) Binomial Distribution Based Tau-Leap Accelerated Stochastic Simulation. *Journal of Chemical Physics*, **122**, 024112. https://doi.org/10.1063/1.1833357

[19] Pettigrew, M.F. and Resat, H. (2007) Multinomial Tau-Leaping Method for Stochastic Kinetic Simulations. *Journal of Chemical Physics*, **126**, 084101-084101-15.
https://doi.org/10.1063/1.2432326

[20] Anderson, D.F. (2008) Incorporating Postleap Checks in Tau-Leaping. *Journal of Chemical Physics*, **128**, 054103. https://doi.org/10.1063/1.2819665

[21] Allen, G.E. and Dytham, C. (2009) An Efficient Method for Stochastic Simulation of Biological Populations in Continuous Time. *BioSystems*, **98**, 37-42.
https://doi.org/10.1016/j.biosystems.2009.07.003

[22] Kermack, W.O. and McKendrick, A.G. (1927) Contributions to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society of London A*, **115**, 700-721.
https://doi.org/10.1098/rspa.1927.0118

[23] Hamilton, M.S. (1980) Estimating Length and Orders of Delays in System Dynamics Models. In: Randers, J., Ed., *Elements of the System Dynamics Method*, the MIT Press, MA, 162-183.

[24] Renshaw, E. (1991) Modelling Biological Populations in Space and Time. Cambridge University Press, Cambridge.
https://doi.org/10.1017/CBO9780511624094

[25] Law, A.M. and Kelton, W.D. (1991) Simulation Modeling and Analysis, McGraw-Hill, New York.

[26] Zeigler, B.P., Praehofer, H. and Kim, T.G. (2000) Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press, San Diego, Ca.

[27] Forrester, J.W. (1961) Industrial Dynamics. MIT Press, Cambridge, MA.

[28] Haefner, J.W. (1996) Modelling Biological Systems: Principles and Applications.

Chapman & Hall, International Thomson Publishing, New York.
https://doi.org/10.1007/978-1-4615-4119-6

[29] Lande, R., Engen, S. and Sæther, B.-E. (2003) Stochastic Population Dynamics in Ecology and Conservation. Oxford University Press, Oxford.
https://doi.org/10.1093/acprof:oso/9780198525257.001.0001

[30] Bailey, N.T.J. (1975) The Mathematical Theory of Infectious Diseases and Its Applications. Griffin, London.

[31] Bratley, P., Fox, B.L. and Schrage, L.E. (1983) A Guide to Simulation. 2nd Edition, Springer-Verlag, New York. https://doi.org/10.1007/978-1-4684-0167-7

[32] Gustafsson, E. (2017) System Dynamics Statistics (SDS): A Statistical Tool for Stochastic System Dynamics Modeling and Simulation (Uppsala). Master Thesis, Uppsala University, IT-17013.
http://uu.diva-portal.org/smash/record.jsf?pid=diva2%3A1095658&dswid=8163#sthash.Jo873tM2.dpbs

[33] Fortmann-Roe, S. (2014) Insight Maker: A General-Purpose Tool for Web-Based Modeling & Simulation. *Simulation Modelling Practice and Theory*, **47**, 28-45.
https://doi.org/10.1016/j.simpat.2014.03.013

[34] Brito, T.B. and Botter, R.C. (2011) A Conceptual Comparison between Discrete and Continuous Simulation to Motivate the Hybrid Simulation Method, In: Jain, S., *et al.*, Eds., *Proceedings of the* 2011 *Winter Simulation Conference*, 11-14 December 2011, Grand Arizona Resor, Phoenix, AZ, 3915-3927.

[35] Özgűn, O. and Barlas, Y. (2011) Discrete vs. Continuous Simulation: When Does It Matter? In: Jain, S., *et al.*, Eds., *Proceedings of the* 2011 *Winter Simulation Conference*, 11-14 December 2011, Grand Arizona Resor, Phoenix, AZ, 22 p.

[36] Powersim AS and Powersim Corporation Staff, Powersim Corporation (1996) Powersim Reference Manual, Powersim Press, Powersim Corporation, 1175 Herndon Parkway, suite 600, Herndon, VA 22170.

[37] Green, G.H. and Donovan, J.W. (1970) The Natural History of Cervical Carcinoma *in situ*. *Journal of Obstetrics and Gynaecology of the British Commonwealth*, **77**, 1-9. https://doi.org/10.1111/j.1471-0528.1970.tb03401.x

[38] Gustafsson, L. and Adami, H.-O. (1989) Natural History of Cervical Neoplasia: Consistent Results Obtained by an Identification Technique. *British Journal of Cancer*, **60**, 132-141. https://doi.org/10.1038/bjc.1989.236

[39] Gustafsson, L. and Adami, H.-O. (1990) Cytologic Screening for the Uterine Cervix in Sweden Evaluated by Identification and Simulations. *British Journal of Cancer*, **61**, 903-908. https://doi.org/10.1038/bjc.1990.202

[40] Paul, C. (1988) The New Zealand Cervical Cancer Study: Could It Happen Again? *British Medical Journal*, **297**, 533-539. https://doi.org/10.1136/bmj.297.6647.533

[41] Gross, D. and Harris, C.M. (1998) Fundamentals of Queueing Theory. John Wiley & Sons, Inc., New York.

[42] Volterra, V. (1926) Fluctuations in the Abundance of a Species Considered Mathematically. *Nature*, **118**, 558-560. https://doi.org/10.1038/118558a0

[43] Luenberger, D.G. (1979) Introduction to Dynamic Systems. Theory, Models and Applications. John Wiley & Sons, Inc., New York.

Appendix A: Illustrative Model Examples

To make the paper as compact as possible, six key examples are presented in this appendix and referred to at several points in main text. These models are presented below as model code and in some cases also as Forrester diagrams [27] [36]. The examples are also accessible for downloading and execution, see **Appendix B**.

Examples:

A1. Consequences of creating stochastic transitions by adding noise.

A2. Continuous modelling of discrete objects hides behaviours and may distort results.

A3. Tragic consequences of neglecting stage-to-compartment expansion.

A4. Ridiculous results from a continuous SIR model and the importance of correct stage-to-compartment expansion.

A5. Queues and statistical devices in CSS.

A6. Combined continuous and discrete modelling in CSS.

Example A1: Consequences of creating stochastic transitions by adding noise

Trying to make a deterministic CSS stochastic by adding or multiplying noise to the transitions is generally not a good idea. Here is a simple example of what can happen.

Radioactive decay, where $X(t)$ represents the number of non-decayed radioactive atoms, with a decay fraction a per time unit can be implemented in a deterministic model as: $X(t + \Delta t) = X(t) - \Delta t \cdot a \cdot X(t)$. If we try to describe the irregular nature of radioactive decay by adding random perturbations to the transition as:

$$X(t + \Delta t) = X(t) - \Delta t \cdot a \cdot X(t) + b \cdot e(t) \tag{6}$$

where $e(t)$ is a zero-mean discrete-time white noise, here assumed Gaussian, we would obtain results as in **Figure A1**.

As illustrated by **Figure A1**, the stochastic model could produce a number of artifacts, such as:

1) Non-integer numbers of the remaining non-decayed atoms.

2) Stochastic variations unrelated to the remaining number of atoms.

3) Sudden increases in the number of atoms.

4) Continued variations, even when all atoms are decayed.

5) A negative number of atoms may occur.

6) The model's behaviour will strongly depend on the time-step used.

The artifacts in this example may seem obvious, but when part of a larger model, they may generate severe consequences that may be hard to trace back to their root cause. Variations without appropriate reasons may excite other parts of the model. A negative number of tokens may trigger various phenomena, e.g. driving other processes backwards. Furthermore, if e.g. the logarithm or the root of the number of remaining atoms is used, the model will crash when this
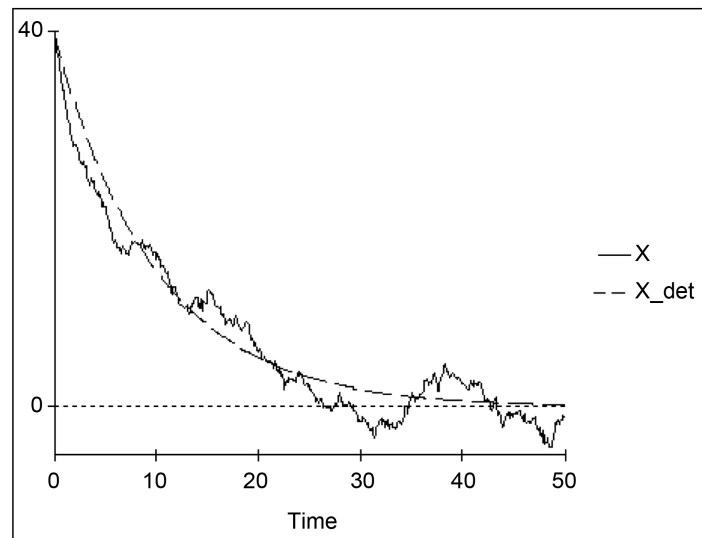
Figure A1. Adding noise to a deterministic CCS model may create a number of artifacts. The results from a deterministic model with added noise (solid line) and without (dashed line) are shown. ($X(0) = 40$, $a = 0.1$ and $b = 4$ where $\Delta t = 0.1$).

number becomes negative.

The reason for failure in this example is that transition stochasticity is an intrinsic part of the dynamic process, and in this case the model should be:

$$X\left(t+\Delta t\right) = X\left(t\right) - Poisson\left[\Delta t \cdot a \cdot X\left(t\right)\right] \tag{7}$$

Then, for a sufficiently small Δt, all the artifacts will vanish. ∎

Example A2: Continuous modelling of discrete objects hides behaviours and may distort results

Continuous modelling removes many of the characteristics of a discrete system because the transition stochasticities will vanish. Here we demonstrate that three fundamentally different discrete models can behave identically when modelled as continuous.

1) First, consider a *discrete logistic growth model* with a population size of $X(t)$:

$$X\left(t+\Delta t\right) = X\left(t\right) + Po\left[\Delta t \cdot a \cdot X\left(t\right)\right] - Po\left[\Delta t \cdot b \cdot X^2\left(t\right)\right] \tag{8}$$

This is an *open, first-order non-linear model*. The model structure and its behaviours from three replications are shown in **Figure A2**. The population continues to fluctuate over time with large variations (with the risk of extinction).

2) Next, consider a *discrete* epidemic *SI model* of $N = S + I$ individuals, where a susceptible set (S) is infected by encounters with members of an infectious set (I). The risk of infection per time unit is proportional to the number of susceptibles, to the number of infectious subjects and to a constant c. Thus:

$$S\left(t+\Delta t\right) = S\left(t\right) - \Delta t \cdot F\left(t\right) \tag{9a}$$
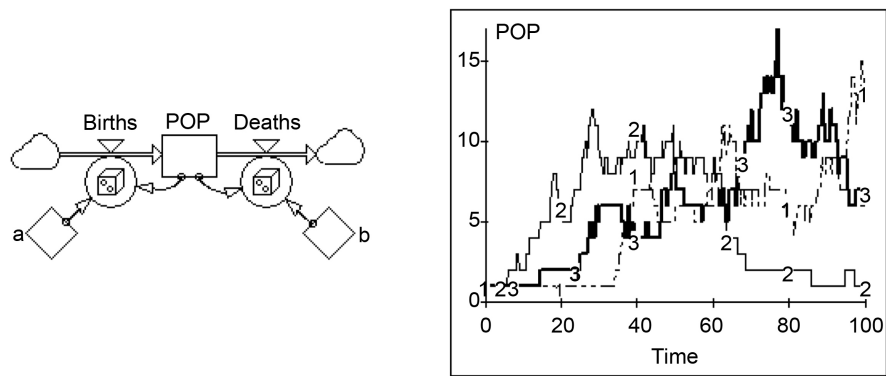
Figure A2. Forrester diagram of a stochastic logistic model and the results of three replications of the model for the case $X(0) = 1$, $a = 0.1$ and $b = 0.01$. In one of the replications, X became extinct.

$$I(t + \Delta t) = I(t) + \Delta t \cdot F(t) \qquad (9b)$$

$$\Delta t \cdot F(t) = Po\left[\Delta t \cdot c \cdot S(t) \cdot I(t)\right] \qquad (9c)$$

This is a *closed, second-order bilinear model*, shown in **Figure A3**, where the number of infectious subjects increases stepwise to $N = S + I$, where it remains.

3) Finally, consider a discrete "pruned prey-predator" model where prey births and predator deaths are removed.

$$PREY(t + \Delta t) = PREY(t) - \Delta t \cdot Deaths(t) \qquad (10a)$$

$$PRED(t + \Delta t) = PRED(t) + \Delta t \cdot Births(t) \qquad (10b)$$

$$Encounters(t) = PREY(t) \cdot PRED(t) \qquad (10c)$$

$$\Delta t \cdot Deaths(t) = Po\left[d \cdot \Delta t \cdot Encounters(t)\right] \qquad (10d)$$

$$\Delta t \cdot Births(t) = Po\left[e \cdot \Delta t \cdot Encounters(t)\right] \qquad (10e)$$

This model consists of two *open and coupled first-order submodels*, shown in **Figure A4**, where the number of predators increases stepwise to an undefined final level (while the prey vanish).

Now, we eliminate the transition stochasticity from the three models by removing the *Po*[ ] clauses. We then obtain the *corresponding* deterministic models that are *embedded* in the stochastic models.

In **Figure A5**, the three deterministic models and their identical behaviours are shown.

As seen in **Figure A5**, the continuous approach removes all the differences between the behaviours of the three models by eliminating the variability created by transition stochasticity. It also introduces bias in the results (not shown here). Finally, it does not capture the risk of extinction in model A if, for example, the first event is death of the only subject initialised at time zero. ∎

## Example A3: Tragic consequences of neglecting stage-to-compartment expansion

This simple but real example cost lives because the modeller did not understand
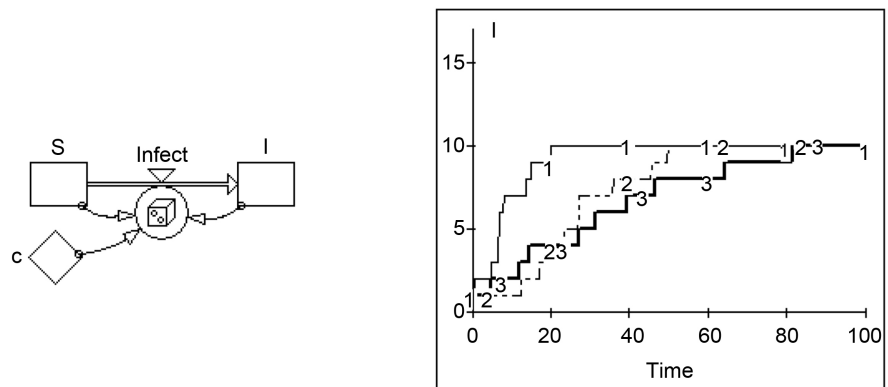
Figure A3. Forrester diagram of the stochastic SI model and three replications of its behaviour. (Results for $S(0) = 9$, $I(0) = 1$ *and* $c = 0.01$.)
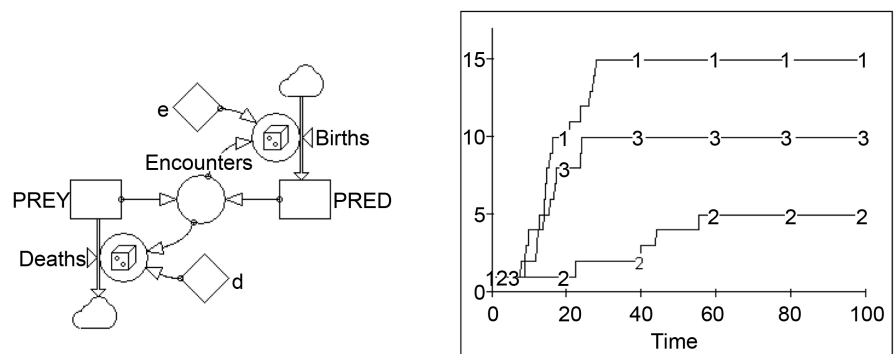
Figure A4. Forrester diagram of the stochastic pruned prey-predator model and three replications showing possible model behaviours of the predators. (Results for PREY(0) = 9, PRED(0) = 1 and $d = e = 0.01$).
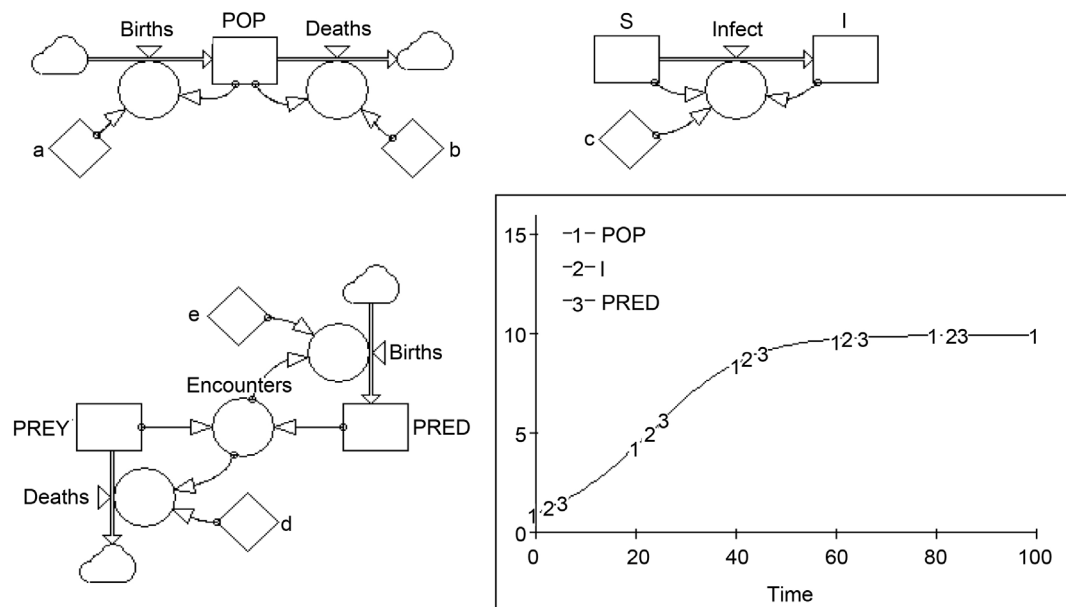
Figure A5. Forrester diagram of the deterministic logistic model, the deterministic SI model and the deterministic pruned prey-predator model, with the same parameter values and initial conditions as in the stochastic cases. The behaviours of these deterministic models are identical.

the difference between a stage and a compartment.

Does cervical cancer develop from cancer *in situ*?

During the 1960s, G.H. Green, a medical professor in Auckland, New Zealand, monitored a large number of women with *cancer in situ* (CIS) of the cervix uteri to determine whether CIS would progress into *invasive* cancer.

His main scientific argument came from a clinical study of 75 women who showed persistent CIS. These women were followed up during 13 to 141 months, with a mean of 53 months (4.42 years). *None of these women developed invasive cancer during the follow-up period* [37]. Interpreting the results from the clinical study by an inappropriate model led Professor Green to dismiss the common opinion that CIS is a pre-stage of cancer and to question the need for any treatment at all of CIS.

In statistical terms, the objective was to test whether the null hypothesis (H_0) that: *CIS will progress to invasive cancer* could be rejected at a high level of confidence based on the observations of the clinical study of 75 women with CIS.

Conceptual model

The conceptual description quoted here is taken from [37], but we have changed the denotation of the sojourn time from m to D.

"ANALYSIS

… it was supposed that every patient developing carcinoma *in situ* had the same probability q of ultimately developing invasive cancer.…

Model 1. *Poisson process*. Suppose that invasion is determined by some chance event which has a constant likelihood with time, regardless of age and the duration of carcinoma *in situ*. …Suppose further that in patients who will not ultimately develop invasive cancer the carcinoma *in situ* heals spontaneously with the same time course …Then if the mean latent period before invasion or regression is in years, the probability that the ith patient followed for T_i years does not develop cancer is
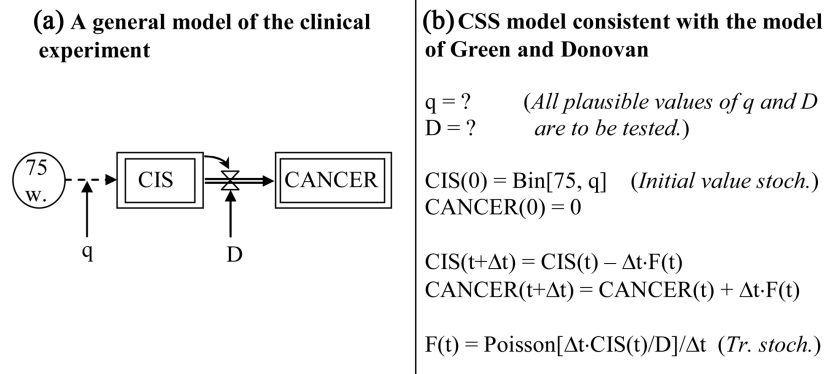
$$1 - q\left(1 - \mathrm{e}^{T_i/D}\right) \qquad (1)$$

With this model we can avoid the need to determine whether any lesions have regressed spontaneously—and hence the need to define criteria for this event. The probability that each patient will ultimately revert to normal is ($1 - q$). …

The probability P that no patients develop cancer is then given by the product of *75* such terms, and has been computed for various values of q and D."

Unfortunately, the cancer *in situ* stage was assumed to have an exponential sojourn-time distribution (see Equation (1) in the quote above). Then, it does not matter whether the conceptual model is realised as a DES, CSS or analytical model. The error is already in place. In CSS model terms, this means that the cancer *in situ* stage is modelled by a single compartment.

A model to interpret the clinical experiment, where the sojourn-time distribution in the cancer *in situ* (CIS) stage is unspecified, is shown in **Figure A6(a)**, and the model equivalent to Green's and Donovan's model, where the CIS-stage is represented by a single compartment, is shown in **Figure A6(b)**.

(a) A general model of the clinical experiment	(b) CSS model consistent with the model of Green and Donovan

$q = ?$ (*All plausible values of q and D*
$D = ?$ *are to be tested.*)

$CIS(0) = Bin[75, q]$ (*Initial value stoch.*)
$CANCER(0) = 0$

$CIS(t+\Delta t) = CIS(t) - \Delta t \cdot F(t)$
$CANCER(t+\Delta t) = CANCER(t) + \Delta t \cdot F(t)$

$F(t) = Poisson[\Delta t \cdot CIS(t)/D]/\Delta t$ (*Tr. stoch.*)

Figure A6. CSS model where *q* out of 75 studied women with a *CIS* lesion [initial value stochasticity: Bin(75, *q*)] are assumed to ultimately develop invasive cancer. Only these women are initiated into the *CIS*-stage. The progressing women are assumed to have an average sojourn time of *D* years in the *CIS*-stage before progressing to invasive cancer. What then is the probability that none of the women will have cancer during the average 4.42 year period of follow-up? (The invasive CANCER stage is only required as a counter.) **Figure A6(a)** shows a general model of the clinical experiment with the *CIS*-stage unspecified, and **Figure A6(b)** shows the corresponding CSS code to Green's and Donovan's model, where the *CIS*-stage is represented by a single compartment.

The interpretation of the clinical experiment as: "… it was supposed that every patient developing carcinoma *in situ* had the same probability *q* of ultimately developing invasive cancer" means that only the women that would eventually end up with cancer had to be included in the model. This number is of course unknown for each new experiment, which is why this uncertainty must be reflected in the model. This is accomplished by initial value stochasticity as: $CIS(0) =$ Bin[75, q].

The very limited objective, of only calculating the probability of obtaining *zero invasive cases* during the follow-up period (which had been observed in the clinical study), given the values of the risk parameters *q* and sojourn time *D*, made it possible to use the very simple analytical model quoted above, although we discuss it in terms of an identical CSS model.

With their model, Green and Donovan rejected the null hypothesis (H_0) that: *CIS will progress to invasive cancer* at a high level of confidence for all plausible combinations of values for *q* and *D*.

Alternative models

What would have happened with realistic modelling of the disease process, i.e. with a realistic sojourn-time distribution for the CIS-stage?

To illustrate the importance of this question, we consider the sojourn-time distributions for the *CIS*-stage by testing different numbers of serial compartments, but with the same overall (and realistic) time parameter value of $D = 11$ years.

The sojourn-time distributions for $k = 1, 3, 5$ and 10 serial compartments (*k*-Erlang(*D*/*k*))b for the *CIS*-stage are shown in **Figure A7(a)** and their cumulative
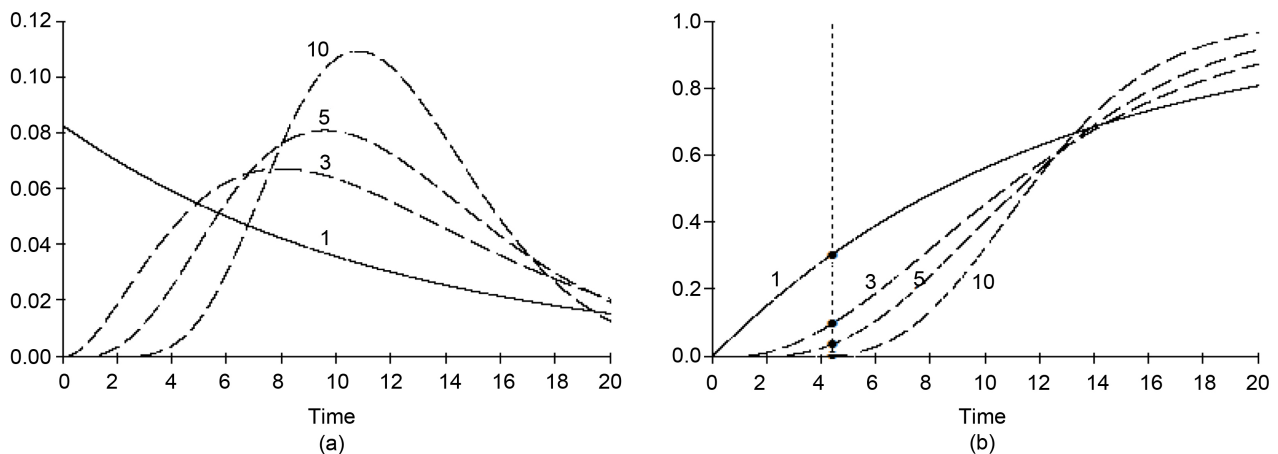
Figure A7. (a) The sojourn-time distributions of one progressing woman developing invasive cancer for $k = 1, 3, 5$ and 10 serial compartments describing the *CIS*-stage with the same total value of the parameter $D = 11$ years; (b) The corresponding cumulative distributions. The average follow-up period of FU = 4.42 years, is marked by a vertical dotted line. Note the huge difference in risk, according to the model, of having cancer after 4.42 years when the *CIS*-stage is represented by 1, 3, 5 or 10 compartments, although the average sojourn time D is the same!

distributions are shown in **Figure A7(b)**.

The follow-up time varied for the 75 women in the study, which is important to recognise when the *CIS*-stage is modelled by more than one compartment, because then it is also important to know how long each woman has already been in the *CIS*-stage at "time zero". In this example we ignore this (using the average follow-up time for all women) in order to only illustrate the consequences of different distributions of the sojourn-time.

For different sojourn-time distributions of the *CIS*-stage, where $k = 1, 3, 5$ and 10 serial compartments are used, the probabilities of *none* of the *75* women developing invasive cancer during the follow-up period are shown in **Table A1**.[6]

Professor Green's model, where the *CIS*-stage is represented by a single compartment, predicts with more than 95% probability that at least one of the women in the clinical study would get cancer. Therefore, he concluded that the opinion that CIS leads to cancer could be rejected with 95% confidence.

However, the null hypothesis that CIS will develop into invasive cancer (with these parameter values) can be rejected at 95% confidence only when using the CIS representation of a single compartment. For $k > 1$, there is no support for rejection of the null hypothesis.

Consequences

Since Green and Donovan failed to understand the importance of modelling the sojourn-time distribution, they rejected the true null hypothesis that cancer *in situ* is a pre-stage to invasive cancer. The consequences of the study was that the professor allowed women with cancer *in situ* to remain untreated during the following decade, which in turn led to a number of cancers and deaths and finally to a legal process where the professor was convicted [40]. ■

[6]In [38] and [39] it was found that the *CIS*-stage must be modelled by *at least* three serial compartments (perhaps more) to obtain results consistent with observations.

Table A1. Probability of obtaining at least one invasive cancer case from 75 CIS cases when $q = 0.12$, $D = 11$ years, and *the follow-up time* = 4.42 years, when the *CIS*-stage is represented by $k = 1$, 3, 5 and 10 serial compartments.

Model	1 compartment	3 compartments	5 compartments	10 compartments
Prob (≥ 1 case out of 75)	95.2%	66.8%	38.2%	7.2%

Example A4: Ridiculous results from a continuous SIR model and the importance of correct stage to-compartment expansion

Here we calculate *the size of an epidemic* obtained from SIR models having the same sojourn-time but with different distributions, and also compare the results from continuous and discrete SIR modelling.

Conceptual model: The model has the following setting: The studied population $N = S + I + R$ initially consists of 1000 susceptible individuals, a single individual that has just entered the infectious stage, and zero recovered individuals. Every individual of the population meets every other under equal conditions in each time unit. When a susceptible individual meets an infectious one, the probability per time unit of the susceptible becoming infected is $p = 0.0003$, and the expected sojourn time in the infectious compartment is $D = 4$ time units.

Figure A8 shows such a SIR model where k serial compartments representing the I-stage generate a k-Erlang(D/k) distribution for the sojourn-time distribution in this stage.

We denote this model SI_kR, where k is the number of serial compartments in the I-stage. (The S-stage is just a reservoir of susceptibles and the R-stage is just a "counter", so they are represented by single compartments.) We also compare the deterministic and stochastic versions of these models.

The model code for a stochastic SI_kR model is:

$$S(t + \Delta t) = S(t) - \Delta t \cdot F1(t) \tag{11a}$$

$$I_1(t + \Delta t) = I_1(t) + \Delta t \cdot F1(t) - \Delta t \cdot F2_1(t) \tag{11b}$$

$$I_2(t + \Delta t) = I_2(t) + \Delta t \cdot F2_1(t) - \Delta t \cdot F2_2(t) \tag{11c}$$

$$\ldots$$

$$I_k(t + \Delta t) = I_k(t) + \Delta t \cdot F2_{k-1}(t) - \Delta t \cdot F2_k(t) \tag{11d}$$

$$R(t + \Delta t) = R(t) + \Delta t \cdot F2_k(t) \tag{11e}$$

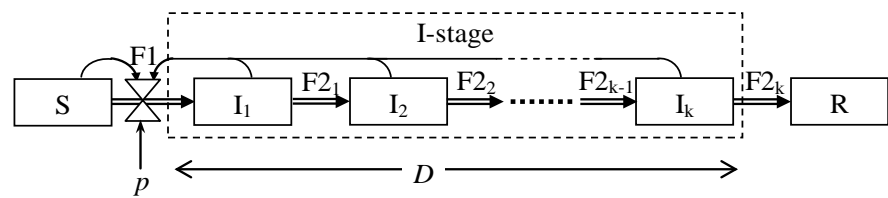$$\Delta t \cdot F1(t) = Po\left[\Delta t \cdot p \cdot S(t) \cdot \left(I_1(t) + I_2(t) + \cdots + I_k(t)\right)\right] \tag{11f}$$

$$\Delta t \cdot F2_1(t) = Po\left[\Delta t \cdot I_1(t) / (D/k)\right] \tag{11g}$$

$$\Delta t \cdot F2_2(t) = Po\left[\Delta t \cdot I_2(t) / (D/k)\right] \tag{11h}$$

$$\ldots$$

$$\Delta t \cdot F2_k(t) = Po\left[\Delta t \cdot I_k(t) / (D/k)\right] \tag{11i}$$

The objective is to study the number of susceptible individuals being infected,

Figure A8. SI_kR model of an infectious disease with transmission parameter p and average sojourn-time in the I-stage D.

S(0) − S(*End*), for $k = 1, 3, 5, 10$ and ∞ for both continuous and a discrete models.

Results: The full results from 10,000 replications of a stochastic model are given by a probability distribution function of the number of susceptibles that become infected. **Figure A9** shows the results for the SI_1R model.

From the 10,000 replications, the average size of the epidemic and its standard deviation are estimated. From this, we also calculate the 95% confidence interval, see **Table A2**.

The average size of the epidemic was 53, 76, 83, 84 and 96 individuals for $k = 1, 3, 5, 10$ and ∞ compartments, respectively. Thus even going from the model of $k = 1$ to $k = 3$ increases the result by more than 40%.

Table A2 also shows that an embedded *deterministic* SI_kR model using the same parameter values produces a fundamentally different outcome than a stochastic model. More surprising, the size of the epidemic estimated by a deterministic SIR model is independent of the number of serial compartments, k, of the infectious stage.

The reason why the deterministic model distorts the results is that it neglects the possibility that the single infectious individual may recover before infecting another susceptible, so that no epidemic will occur. ∎

Example A5: Queues and statistical devices in CSS

Queues, which are central in DES, can be smoothly modelled within CSS. Queuing estimates such as average number of tokens waiting, average waiting time, utilization of a resource (that can be busy or idle) can then easily be calculated within a CSS model.

Implementation of queuing systems in CSS is straightforward. This is demonstrated for the $M/M/1$, $M/M/s$ and $M/M/\infty$ cases, where M stands for "Markov" or "Memoryless" which means exponentially distributed inter-arrivals and service times, and the third symbol represents the number of parallel servers, see [41].

Let QS denote the actual number of *queuing and served* tokens represented by a compartment with the same name. The input is a Poisson process, with the average arrival rate, λ, and the customers are served by s servers working in parallel, with the average service rate, μ, for each server. This gives the stochastic model:

$$QS(t + \Delta t) = QS(t) + \Delta t \cdot In(t) - \Delta t \cdot Out(t) \tag{12a}$$
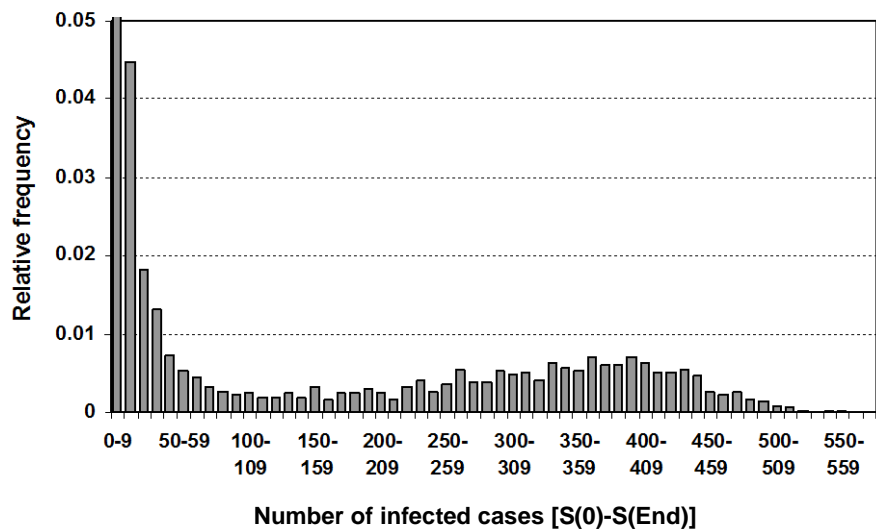
Figure A9. A p.d.f. of the number of susceptible individuals becoming infected [$S(0)–S(End)$] calculated from 10,000 replications of the stochastic SI_1R-model. The bar intervals are 0 - 9, 10 - 19, ⋯, 550 - 559. (The first bar (0 - 9) is truncated. It has the value 0.7422).

Table A2. Number of individuals contracting the disease. Results obtained from 10,000 replications of the *stochastic* models and from the corresponding *deterministic* models.

	SI_1R	SI_3R	SI_5R	$SI_{10}R$	$SI_\infty R$
Stochastic model: Average	53.1	75.7	83.0	84.1	96.2[a]
(95% C.I.)	(50.7 - 55.5)	(73.0 - 78.4)	(80.2 - 85.8)	(81.3 - 86.8)	(93.3 - 99.1)
Deterministic model	318.5	318.5	318.5	318.5	-

[a]To realise $k = \infty$ (which means a constant time-delay of exactly D time units), a stochastic DES model with the fixed sojourn time of D time units was used.

$$In = Po\left[\Delta t \cdot \lambda\right]/\Delta t \quad \text{(We drop “(t)” for simplicity)} \tag{12b}$$

$$Out = Po\left[\Delta t \cdot \mu \cdot MIN(QS,1)\right]/\Delta t \quad \text{for the } M/M/1 \text{ case;} \tag{12c}$$

$$Out = Po\left[\Delta t \cdot \mu \cdot MIN(QS,s)\right]/\Delta t \quad \text{for the } M/M/s \text{ case;} \tag{12d}$$

$$Out = Po\left[\Delta t \cdot \mu \cdot QS\right]/\Delta t \quad \text{for the } M/M/\infty \text{ case (“self-service”)} \tag{12e}$$

Remarks: $MIN(QS, s)$ means that at most s tokens can be served simultaneously, and if $QS < s$ only QS tokens are served. Of course, λ, μ and also the number of available servers s can be varied during the simulation. For further discussion on queuing models in CSS, see [8].

Figure A10 shows an M/M/s queue model with a number of devices for internal statistics. The devices to study the internal statistics of the queue in **Figure A10** are:

In_Count and *Out_Count* count the customers that enter and leave the queue. *Cum_QS_Time* accumulates *QS* over time to obtain the total waiting time in the compartment *QS*. Dividing *Cum_QS_Time* by *In_Count* gives *Av_QS_Time*,
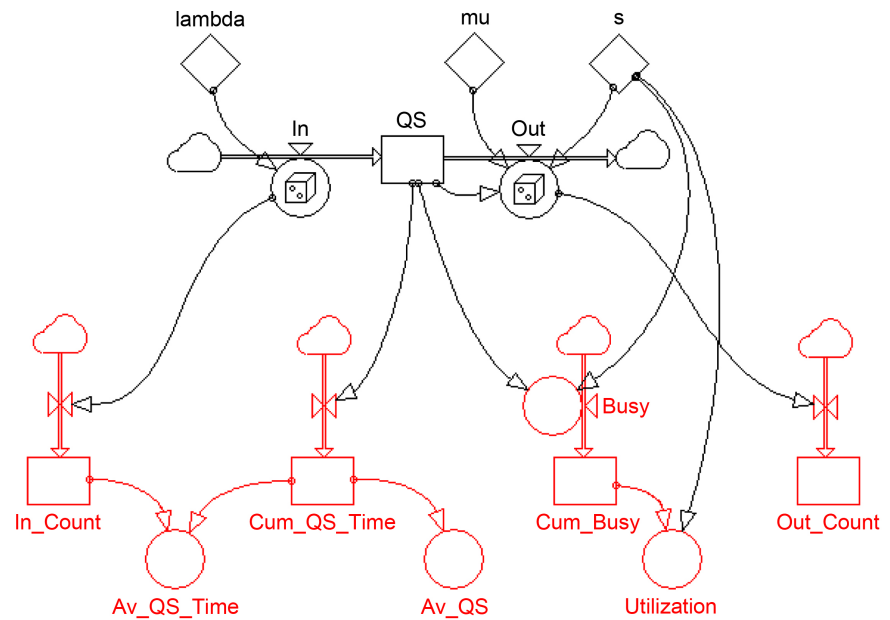
Figure A10. An *M*/*M*/*s* queuing system. Here, *QS* is a compartment holding the total number of currently queuing and served tokens. The arrival rate is *λ* (*lambda*), the service rate (per server) is *μ* (*mu*) and the number of parallel servers is *s*. The lower part of the diagram shows devices for counting arrivals and departures, and for calculating average queue-time, queue-length and server utilization.

and dividing it by *Time* gives *Av_QS* (the average number in the queuing system).

Finally: *Busy = If QS < s then QS else s*, which is accumulated over time in *Cum_Busy*. The (average) *Utilization* of the *s* servers over the replication is then: *Cum_Busy*/(*s*Time*).

Because the tokens have no identity, they cannot be followed, so min and max through-times cannot be obtained. ■

Example A6: Combined continuous and discrete modelling in CSS

Conceptual model: The Lotka-Volterra equations describe a prey-predator system for two species *X* and *Y* by differential equations [42] [43]. The prey breeds at a rate proportional to its size *X*, and is reduced because of "encounters" with predators, which are proportional to *X·Y*. We also include competition among prey, proportional to X^2. The "encounters" with prey give the predators the energy to breed, so they increase in proportion to *X·Y*. Finally, the predators die in proportion to their number *Y*. The continuous Lotka-Volterra model then has the mathematical form:

$$\mathrm{d}X/\mathrm{d}t = aX - bXY - cX^2 \tag{13a}$$

$$\mathrm{d}Y/\mathrm{d}t = dXY - eY \tag{13b}$$

where *a* and *d* are fertility constants, *b* and *e* mortality constants, and *c* is a proportionality constant for competition. In our setting, we let the prey be a *continuous amount of biomass X* (e.g. of grass), while the predators *Y* are a *discrete number of tokens* (e.g. sheep). The combined continuous-discrete model can

then be rewritten as:

$$X(t + \Delta t) = X(t) + \Delta t \cdot F_1 - \Delta t \cdot F_2 - \Delta t \cdot F_3 \quad \text{(Continuous grass)} \tag{14a}$$

$$\Delta t \cdot F_1 = \Delta t \cdot a \cdot X(t) \tag{14b}$$

$$\Delta t \cdot F_2 = \Delta t \cdot b \cdot X(t) \cdot Y(t) \tag{14c}$$

$$\Delta t \cdot F_3 = \Delta t \cdot c \cdot X^2(t) \tag{14d}$$

$$Y(t + \Delta t) = Y(t) + \Delta t \cdot F_4 - \Delta t \cdot F_5 \quad \text{(Discrete sheep)} \tag{14e}$$

$$\Delta t \cdot F_4 = Po\left[\Delta t \cdot d \cdot X(t) \cdot Y(t)\right] \tag{14f}$$

$$\Delta t \cdot F_5 = Po\left[\Delta t \cdot e \cdot Y(t)\right] \tag{14g}$$

The behaviour of the combined continuous-discrete model, starting at the equilibrium $X(0) = 60$ tons of grass and $Y(0) = 28$ sheep with $a = 0.2$, $b = 0.005$, $c = 0.001$, $d = 0.005$ and $e = 0.3$, is exemplified in Figure A11.

In the replication shown in Figure A11, the discrete predators become extinct at around 450 time units, so the continuous prey increases logistically and now without irregular variations.

Comparing the continuous model (13) with the combined model (14) reveals that the continuous model rapidly reaches an equilibrium state for both species without further variations. Starting at the equilibrium state used here only produces the two straight lines: $X(t) = 60$ and $Y(t) = 28$. Furthermore, a phenomenon such as extinction cannot occur for model (13). ∎

Appendix B: Supplementary Material

We want to provide the following means to the reader:

1) To hands-on try stochastic CSS and to test and see the differences between stochastic and deterministic CSS model behaviours, the six examples A1 to A6 presented in Appendix A are made available from internet.

2) A tool for statistical post-analysis of multiple runs of a stochastic CSS model is also provided there.

B1. Examples A1 to A6
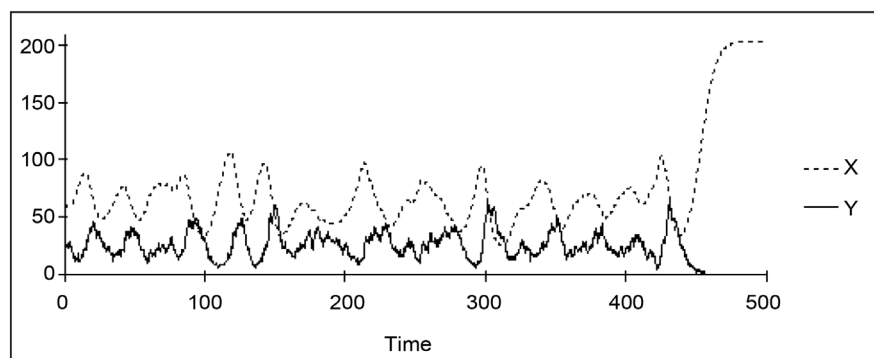
Figure A11. A replication of the prey-predator model with continuous prey (*X*) and discrete predators (*Y*) using the combined model (14).

In order to give the reader a deeper understanding and experience of stochastic CSS modelling, we provide a simple an easy-to-use stochastic CSS language called **StochSD**, see Section B2. The frequently referred six models, A1 to A6 in Appendix A can be downloaded and executed in StochSD. Alternatively, you can have these six models in Insight Maker.

To try Stochastic CSS go to: https://stochsd.sourceforge.io/homepage/. From here you can get access to the models A1 to A6 in both StochSD and in Insight Maker.

If preferred, you can build and simulate stochastic CSS models in any CSS package where a Poisson distributed random number generator is included.

B2. StochSD including the StatRes software for statistical post-analysis

StochSD (Stochastic System Dynamics) is a rudimentary open-source tool (where the graphics is not yet very elegant) for deterministic and stochastic modelling and simulation based on Insight Maker[7] version 5. StochSD includes an advanced and well tested tool **StatRes** for statistical post-analysis and some other tools. StochSD is open-source, which means that you get it for free. It also gives you access to the source code and the right to modify the code under the open-source licence. StochSD is written in HTML and JavaScript.

StatRes (Statistical Results) performs output analysis of a stochastic model written in StochSD. It works on an opened StochSD model file, orders N replications, collects results of specified quantities and calculates statistics of averages, standard deviations, confidence intervals, min & max values and percentiles. It can also present the statistics in the form of histograms, box plots and scatter plots (X-Y plots) with coefficient of correlation between X and Y calculated, etc.

For using StochSD (including StatRes) over the net or downloading it, go to: https://stochsd.sourceforge.io/homepage/. Here you also find detailed documentation of StochSD and StatRes.

System requirements: Any operative system that can run a modern web-browser, including Windows 7 or later versions, Mac OS X or GNU/Linux. The web-browsers that are well tested for StochSD are Google Chrome and Mozilla Firefox. (It does not work on Internet Explorer.) The web-browser will load StochSD and run your models.

It is our hope that other researchers or CSS software developers will help to make stochastic CSS powerful by including appropriate tools for statistical post-analysis to various CSS packages. We therefore provide open code for **StochSD** and its included tool **StatRes** in particular.

---

[7]We have used the open System Dynamics part: simulation engine, function library, error detection facility, macro facility etc. from Insight Maker. However, the non-open parts based on mxGraph (for model flowchart and result diagrams) and Ext JS (for modelling window and buttons etc.) have been replaced in order to make StochSD completely open. Furthermore, the file handling has been re-written. Finally, we have made some modifications to conform to System Dynamics conventions.