

# A Framework for Constructing Discrete Event Decision Models Using Free, Open-Source Software

Ian Cromwell (Supervisors: Nick Bansback, Stuart Peacock)

Health Economist - ARCC BC Cancer Agency; PhD Candidate – School of Population and Public Health, University of British Columbia

## The Problem

Discrete Event Simulation (DES) models are a powerful way of simulating health care decision problems. They are distinct from more common Markov models, simulating the passage of entities (simulated people) through health and health care events in terms of continuous time rather than fixed cycles. DES models are well-suited to incorporate individual history and consider competing risks, which is more challenging to do in a Markov model.

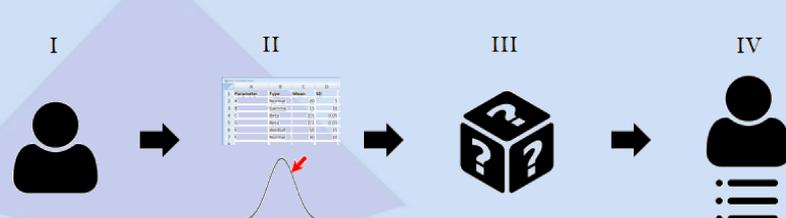
Despite their usefulness, DES models are complicated, and little guidance exists for how to build them. This expertise gap, combined with the high cost of licensing commercial DES software, presents a challenge for scientists and decision makers who wish to build models collaboratively, or share them across health care jurisdictions. Peer review, dissemination, and replication of DES models is challenging for similar reasons.

## The Framework

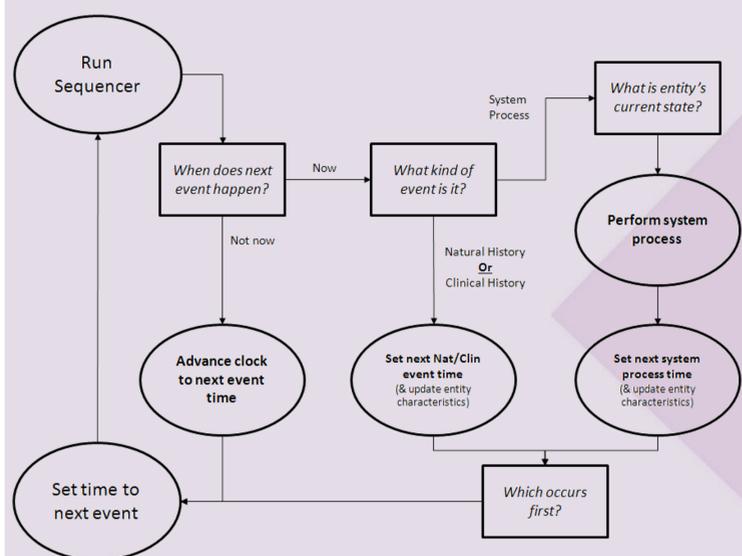
The proposed framework involves the interplay of five basic types of **functions**, and five basic types of **programs**. These work together to create a simulated population of people who experience health status changes, receive health care interventions, and use resources, at discrete points in time over the course of their lives.

## Model Functions

- Entities** are simulated people. They contain model-defined characteristics (e.g., age, disease status, treatment flags, time to next event) that can be used to inform other model functions. Entities take the form of Python 'libraries' that can add, change, or delete variables at any time.
- Parametric sampling** generates necessary values (e.g., probabilities, time to event, unit costs) based on values read in from an Excel spreadsheet. These values typically take the form of a mean, standard deviation, and a value denoting an assumed parametric distribution (Normal, Weibull, Beta, etc.). Parameters can also be input as regression coefficients.
- Probability nodes** are points where an entity's (I) next action is decided by comparing a sampled value (II) to a random number (III). If a given condition is met (e.g., sampled value is larger than the random number), then a model-defined outcome (IV) will transpire (e.g., an entity characteristic changes, an event is scheduled, a treatment is assigned, etc.).
- Time values** – disease events, system process events, clinical history events – are recorded within the entity during the course of the simulation based on its characteristics and sampled values from the spreadsheet. Event times are used to decide the entity's next step, and can be recorded and used to calculate values like QALYs.
- Resources, utilities, and events** are recorded in Python 'lists'. These values can be used to perform unit costing and calculate QALYs, among other uses.



## Model Programs



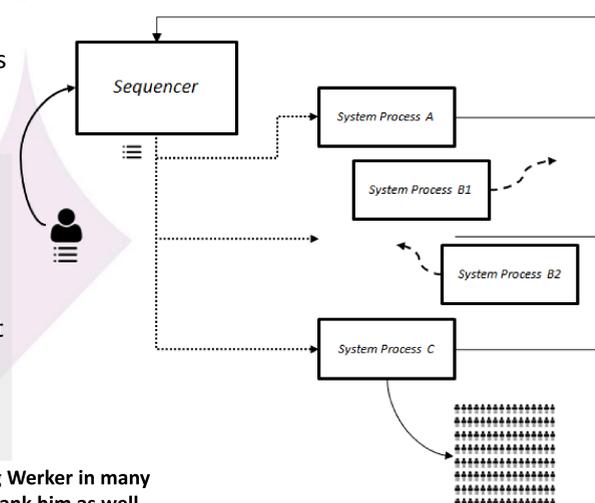
- The Sequencer** is a top-level program that reads the time, position, and characteristics of the entity and runs the appropriate set of **functions** in response.
- The Clock** identifies the current system time and schedules the next event to occur.
- Natural History** describes an entity's trajectory through different disease events in the absence of any intervening treatment (e.g., when they first get sick, when disease becomes symptomatic, etc.).
- Clinical History** describes an entity's trajectory through different disease events after diagnosis/treatment (e.g., time to recurrence, time to remission, time to death from disease).
- System Processes** describe an entity's trajectory through the health care system (e.g., time of next treatment, follow-up appointment, screening appointment)

The Sequencer evaluates the location of the entity within the overall model pathway, and the Clock compares the current time to the next event (Natural History, Clinical History or System Process) that is scheduled to occur. The time to next event is noted, and the appropriate program is run to update the entity's characteristics. This process repeats until the entity reaches a terminal condition (i.e., death). Multiple entities can be simulated in this way, creating a hypothetical population from which costs and QALYs can be estimated.

**Decision analysis** can be performed by adjusting the appropriate model parameter values, and/or by inserting new system process **programs** into the model. Mean costs and survival can be compared between simulated populations to calculate ICERS. The probability node and parametric sampling **functions** incorporate first- and second-order uncertainty, allowing the user to perform probabilistic analysis (CEACs).

## Progress To Date

The first implementation of the framework is the subject of my thesis work – a Whole Disease Model (WDM) of cancers of the oral cavity. The model encompasses screening, premalignant management, treatment, recurrence, and follow-up of oral cancers. Parameters were estimated from published literature and retrospective analysis of premalignant and malignant oral cancer case and trial data within BC. The model is currently being used to conduct cost-effectiveness analysis on two distinct decision problems: use of FV-guided surgery for early stage cancer, and genome-guided management of premalignant disease. Results are anticipated by the end of the year. I am very interested in finding further applications for this framework, and open to collaborative work.



I would like to acknowledge the support of my supervisors Dr. Nick Bansback and Dr. Stuart Peacock, as well as the guidance provided by Dr. Greg Werker in many aspects of the code design. Stavros Korokithakis of Stochastic Technologies was instrumental in most aspects of the code design, and I'd like to thank him as well.