ScienceDirect

Contents lists available at sciencedirect.com
Journal homepage: www.elsevier.com/locate/jval

Methodology

Leveraging DICE (Discretely-Integrated Condition Event) Simulation to Simplify the Design and Implementation of Hybrid Models

J. Jaime Caro, MDCM,* Ross Maconachie, MSc, Michael Woods, MSc, Bhash Naidoo, PhD, Alistair McGuire, PhD

ABSTRACT

Objectives: Using an example of an existing model constructed by the National Institute for Health and Care Excellence (NICE) to inform a real health technology assessment, this study seeks to demonstrate how a discretely integrated condition event (DICE) simulation can improve the implementation of Markov models.

Methods: Using the technical report and spreadsheet, the original model was translated to a standard DICE simulation without making any changes to the design. All original analyses were repeated and the results were compared. Aspects that could have improved the original design were then considered.

Results: The original model consisted of 32 copies (8 risk strata × 4 treatments) of the Markov structure, containing more than 6000 Microsoft Excel® formulas (18 MB files). Three aspects (nonadherence, scheduled treatment stop, and end of fracture risk) were handled by incorporating weighted averages into the cycle-specific calculations. The DICE implementation used 3 conditions to represent the states and a single transition event to apply the probabilities; 3 additional events processed the special aspects, and profiles handled the 8 strata (0.12 MB file). One replication took 16 seconds. The original results were reproduced but extensive additional sensitivity analyses, including structural analyses, were enabled.

Conclusion: Implementing a real Markov model using DICE simulation both preserves the advantages of the approach and expands the available tools, improving transparency and ease of use and review.

Keywords: nonadherence, transition, breast cancer, DICE simulation, hybrid models, Markov, NICE, spreadsheets.

VALUE HEALTH. 2020; 23(8):1049–1055

Introduction

For many health technology assessments (HTAs), it is necessary to design and implement a framework that can take data from various sources and combine it with clinical knowledge and assumptions when necessary to make predictions about the likely health and economic impact of the technologies in question. The cohort state-transition ("Markov") technique remains by far the most common approach to these models, despite its limitations.[1] These Markov models conceptualize the problem in terms of the health states that fractions of the population can be in and the changes ("transitions") in these fractions over time.[2] The transitions are computed repeatedly in periods ("cycles") of time until a time horizon is reached or all of the population is in an end state, referred to as an absorbing state, such as "dead."

Most HTA problems do not fit neatly into the Markov framework because in the course of an illness and its management, various things happen at specific points in time and these occurrences are difficult to represent as states.[3] This is the case for many clinical events and for alterations of treatment. To remain true to the Markov concepts, modelers try to incorporate these into the values (eg, costs, utilities) of the states using weighted averages or by modifying the transition probabilities to reflect, for example, that some people in a state have stopped treatment. These adjustments can greatly increase the complexity of the implementation and make debugging the model and reviewing the final product more difficult. Nevertheless, modelers choose to face these challenges in exchange for the ability to use a familiar method that computes results deterministically using a spreadsheet like Microsoft Excel.

Recently, a new approach to conceptualizing and implementing models was introduced, known as the discretely integrated condition event (DICE) simulation.[4] DICE provides an alternative that can be used to create anything from survival partition models[5] to Markov models to unconstrained discrete event simulations.[6] Indeed, as demonstrated in this exercise, hybrids that take aspects from different techniques are readily implemented. Information in DICE is represented in named "conditions," and changes to that information are specified as expressions in "events" triggered at specific points in time during the simulation.

* Address correspondence to: J. Jaime Caro, MDCM, 39 Bypass Rd, Lincoln, MA, USA, 01773. Email: j.caro@lse.ac.uk or jaime.caro@evidera.com

Both the conditions and the events are tabulated. Execution of a DICE model requires a discrete integrator that reads the tables and, beginning with the Start event, processes each row in the event table. Although time in DICE is continuous (ie, events can happen at any moment), it can be handled discretely to accord with the discrete cyclical approach conventionally used in Markov models.

Implementation in Excel requires that the text expression in each row respects Excel syntax, so a macro can translate it to a formula it can compute. Once all the rows in a table are processed, the macro finds the next event (the one with the lowest scheduled time), processes its table rows, and continues this process until the End event, at which point it stops and reports the contents of those conditions that have been specified as outputs. The DICE macro (83.2 KB Excel macro file) used in this exercise was formulated by J. Moller in Visual Basic for Applications[7]; it is open source and can be downloaded for free from https://www.evidera.com/dice/.

In this article, we report on the first exercise undertaken as part of work package 2 of a Horizon 2020 research and innovation program grant (No. 779312) to develop a user-friendly, validated DICE software and training platform for use by HTA agencies in Europe and elsewhere (see http://impact-hta.eu). The purpose of this exercise was to address important questions that have been raised about use of the DICE approach: Is it just another way of doing discrete event simulation?[8] Is it possible to do Markov models and, if so, why use DICE? Can it address real problems beyond the simplified models used in teaching? Is it feasible to use in terms of execution speed? What does it provide that current approaches do not? In the first section, we describe the problem and the original implementation as a standard Markov model. We then explain how this problem is conceptualized as a DICE model and detail the implementation. In light of this exercise, we discuss the advantages and limitations of using the DICE approach with Markov models.

Methods

To address the questions posed, we asked the guidelines health economic team at the National Institute for Health and Care Excellence (NICE) in England to select an example of a moderately complex HTA problem that had been addressed with a Markov model constructed in MS Excel. By replicating an existing agency model already used to inform real decisions, we aimed to illustrate how DICE simulation can be leveraged to design and implement a hybrid model that respects the state-transition approach but makes it simpler to reflect problem complexity appropriately.

The Decision Problem

The original HTA Markov model was constructed to evaluate the use of chemoprevention in women at moderate to high risk for breast cancer.[9] (NB. to obtain this model a formal request must be made to NICE). This exercise was chosen by the team at NICE because it was complex enough that it would test the theorized benefits of DICE over traditional Markov methods but also simple enough that the re-creation and validation of the model in DICE could be undertaken robustly without substantial effort. Although perhaps other methodologic approaches could have been taken, the model is described here exactly as it was built.

Three agents were compared with no chemoprevention in postmenopausal women stratified into 2 risk groups and 4 age intervals. Annual probabilities of breast cancer without chemoprevention were estimated for each of the 8 strata, and age was used to derive the probability of death from the UK life table. Chemoprevention was assumed to reduce the risk of breast cancer as reflected in an agent-specific relative risk. Three adverse events (AEs) were considered: endometrial cancer, thromboembolism (TE), and fractures (hip, wrist, vertebral, or other). The risks depended only on chemoprevention use and were assumed to be constant over time. Half of the women were assumed to discontinue chemoprevention at 1 year with the immediate loss of all benefits and any adverse events. After 5 years, the remaining women have a planned discontinuation of treatment with no loss of the beneficial effects and an alleviation of the endometrial cancer and TE risks. The fracture risk was assumed to persist for another 5 years. Costs of chemoprevention and required monitoring, and of development of breast cancer, were applied. Utilities were not applied. Instead, the number of quality-adjusted life-years (QALYs) that would need to be gained to meet a given cost-utility threshold were calculated for each strategy.

The Original Model

To enable estimates of the effects of chemoprevention, a 3-state Markov model was used in the original framework. The states consisted of healthy (no breast cancer), breast cancer, and dead. This Markov structure was replicated 32 times (1 for each of the 8 age-risk strata and for each of the 4 chemoprevention strategies). The implementation involved 2 Excel workbooks, 1 for each risk stratum (high and moderate). In each workbook there were 4 worksheets (1 per strategy) where the model calculations were done. Each strategy worksheet was organized into 4 sets of columns where the age-dependent Markov models were implemented. In traditional fashion, the rows represented time in annual cycles, and 3 columns processed the transition probabilities using equations such as:

$$= \mathrm{AR5} * (1 - \mathrm{VLOOKUP}(\mathrm{MIN}(\mathrm{AP5}, 100), '\mathrm{Life\ table}'!\$B\$5{:}\$C\$105, 2, \mathrm{FALSE})) + \mathrm{AQ5} * \mathrm{VLOOKUP}(\mathrm{AP5}, '\mathrm{Life\ table}'!\$F\$5{:}\$K\$130, 2, \mathrm{FALSE})$$

where the first portion computes the number remaining alive using age to look up the cumulative death probability in the life table, and the second portion computes the number transitioning into the state based on the age-dependent annual probabilities (also derived by treatment on the life table worksheet). Three additional columns computed the annual number of new cases of breast cancer and the annual number of each of the 3 AEs. For AEs in the active chemo-prevention models, the formulas used an IF statement to check whether the cycle time was beyond the scheduled stop of chemoprevention to reduce the rates back to those without treatment.

Additional sets of columns on each worksheet created weighted averages using the proportion of the population in each age group. Bone fractures are a side effect of treatment and the model also split the total fractures into the specific bone fracture type to account for the different costs of managing each site. Another set of columns applied a half-cycle correction to each aggregate column by taking the average of the current and next cycle. A final set of 13 columns applied the treatment discontinuation rate at 1 year by using a weighted average of the compliant state memberships for the treatment at issue and no treatment models.

The costs were applied to the state memberships and AE counts in another worksheet consisting of 4 groups of 21 columns, also organized with rows as proxies for time cycles. These were used to summarize the deterministic results on another worksheet. Probabilistic sensitivity analyses were carried out by a

macro and the results were pasted onto a worksheet and processed in other worksheets. Several worksheets contained the inputs. Construction of this model took approximately 50 hours, although a couple of months was spent understanding the decision problem, sourcing parameters, and undertaking preparatory statistical work.

Drawbacks to the Original Model

Several aspects of the original specifications did not fit well into the Markov structure because they are awkward to consider as states. This includes the scheduled treatment stop, nonadherence, and the persistence of fracture risk. Incorporating these made the model more difficult to implement. Because time was in discrete cycles of fixed length, these additional aspects were forced to occur at these time points, even if in reality they might have occurred in between.

Despite its well-designed organization, the implementation may be considered convoluted, requiring 2 workbooks and 13 worksheets, where the Markov structure is repeated 32 times. Although this did not prevent the model from producing results, the increased complexity made it more difficult to comprehend the basic structure, made the tracing of inputs and events onerous, increased the likelihood of error, and set hurdles for reviewers and validators.

The setup of the model means structural sensitivity and scenario analysis cannot be completed without significant additional effort, if at all, for some of the model parameters. Nevertheless, given the strong assumptions that were required, such analyses are necessary to improve understanding of their implications for the results.

In total, the model was more than 17 megabytes, a significant file size for a relatively simple design. Although modern computers can easily accommodate such files, they do take longer to open and execute and may pose problems for transferring across users, such as through email systems.

The DICE Implementation

The original breast cancer chemoprevention Markov model was reconstructed using a DICE simulation (the fully functional DICE model, 122.5 kilobyte Excel file, and DICE macro are downloadable from www.impact-hta.eu). Because a major objective was to explore the advantages of implementing this model using DICE while confirming that this version would replicate exactly the results obtained with the original Markov model[10] but in a simpler manner, no attempt was made to redesign the model; it was translated exactly as originally specified. Apart from the 8 hours spent understanding the original model, construction of the DICE version took 11.5 hours: 4.5 hours were spent implementing the worksheets and 7 hours were for verification and comparison analyses.

The events in the DICE model (Fig. 1) and their initial times are tabulated in the "All Events" table. Note that the names of conditions (here capitalized and italicized) and events (capitalized and bolded) can be anything the modeler likes, provided they are unique; for clarity it is preferable to select descriptive names.

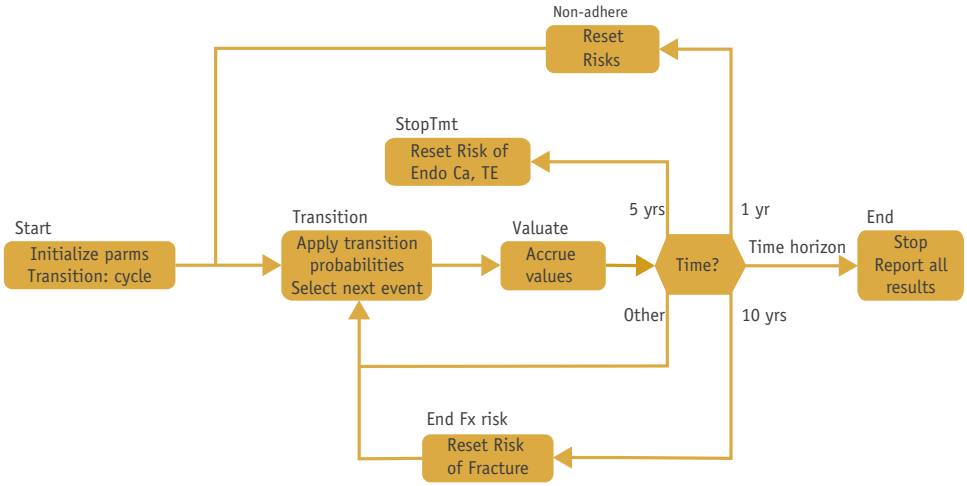
Representing the Markov component

The 3 original states are represented in DICE as conditions: *Healthy*, *BreastCa*, and *Dead*. Each condition contains the state membership, with initial values of 100% *Healthy*, 0% *BreastCa*, and 0% *Dead*. The transitions among these states are applied in a single transition event (Table 1) as the product of the corresponding probability and state membership (3 rows). For example, the number developing breast cancer is given by *Healthy*pBreastCa* and stored in *HealthyBreastCa*. Once these are applied (3 rows in the event table), the new state memberships are collected. For example, *Healthy* is now: *Healthy-HealthyDead-HealthyBreastCa*. A new Transition is scheduled to happen at *Time+Cycle*, and the next event is found by determining the lowest of the current event times: *MIN(CurEventTime)*.

Representing the other aspects

The 3 aspects of the original specifications that did not fit well into the Markov approach (nonadherence to chemoprevention, scheduled treatment stop, and end of fracture risk) are treated in DICE as events. Nonadherence to chemoprevention (Table 2) is scheduled at 1 year. It uses *OnTmt* to represent the proportion of women who continue treatment and applies this in weighted averages to adjust the relevant components, which represent the probabilities, costs, and relative risk (RR) for the new mix of treated/untreated after year 1. To ensure the event does not recur, its next time is set to a very large number (stored in a condition called *Never*).

Figure 1. Basic structure of the original 3-state Markov model. Circles represent states and the arrows represent the possible transitions between them.



DICE indicates discretely integrated condition event; Endo Ca, endometrial cancer; TE, thromboembolism.

Table 1. Tabulated expressions for the transition event.

Type	Name	Expression
Condition	HealthyDead	Healthy*pHealthyToDead
Condition	BreastCaDead	BreastCa*pBreastCaToDead
Condition	HealthyBreastCa	Healthy*pBreastCa
Condition	Dead	Dead + HealthyDead + BreastCaDead
Condition	BreastCa	BreastCa – BreastCaDead + HealthyBreastCa
Condition	Healthy	Healthy - HealthyDead - HealthyBreastCa
Event	Transition	TIME + Cycle
Condition	NextEventTime	Min(CurEventTime)
Condition	NextEvent	Match(NextEventTime, CurEventTime, 0)

The scheduled stop of treatment at 5 years is implemented in an event (Table 3) that sets the treatment-related costs to zero and returns the risks of AEs to the untreated ones, except for the fracture risk, which persists for some additional time. To turn off the increased fracture risk at the required time, EndFractureRisk is scheduled at Time+5, where fracture risk is reset to that without chemoprevention.

The modeling events

Every DICE model needs Start and End events. In this model, End has a single line that updates the simulation clock. Start is somewhat more complicated (Table 4) because it sets up the simulation for each intervention by selecting relevant RR, costs, and AE risks from the corresponding conditions. The CHOOSE statements use *IntervNum*, which the macro automatically increases by 1 after simulating each intervention. Of note, these intervention-specific expressions could also be moved to an InitiateTreatment. This would be particularly useful if there were to be treatment switches during the simulation, thus requiring further treatment initiations.

Another modeling event that can be useful is Valuate (Table 5). This optional event is implemented to group together all the

Table 2. Expressions for the nonadherence event.

Type	Name	Expression
Condition	RRBreastCa	RRBreastCa*OnTmt + (1-OnTmt)
Condition	pEndoCa	pEndoCa*OnTmt + ipEndoCaNoTmt*(1-OnTmt)
Condition	pThromboEmb	pThromboEmb*OnTmt + pThromboEmbNoTmt* (1-OnTmt)
Condition	pFracture	pFracture*OnTmt + pFractureNoTmt*(1-OnTmt)
Condition	CostInterv	CostInterv*OnTmt
Condition	CostMonitor	CostMonitor*OnTmt
Event	NonAdhere	NEVER
Condition	NextEventTime	MIN(CurEventTime)
Condition	NextEvent	MATCH(NextEventTime, CurEventTime, 0)

Table 3. Expressions specifying the stop treatment event.

Type	Name	Expression
Condition	pEndoCa	pEndoCaNoTmt
Condition	pThromboEmb	pThromboEmbNoTmt
Condition	CostInterv	0
Condition	CostMonitor	0
Event	EndFxRisk	TIME + 5
Event	StopTmt	Never
Condition	NextEventTime	MIN(CurEventTime)
Condition	NextEvent	MATCH(NextEventTime, CurEventTime, 0)

expressions that apply values (eg, unit costs, utilities) to accrue outputs, such as QALYs and total costs. By grouping them in 1 event, they are written only once (thus minimizing errors) and can be called from any event as appropriate. For example, treatment cost is accrued as *CostTmt*+*Cycle***CostInterv*, with *CostInterv* set in Start according to current intervention. In the original model, a half-cycle correction was implemented by averaging the number of people alive at the beginning and end of the cycle. In DICE, this is done using the expression AVERAGE(*PriorAlive*, 1-*Dead*) with the result stored in *AliveHalfCycle*. This is used as a multiplier of (*Cycle***UnitValue*). *PriorAlive* is set in Transition before updating *Dead* state membership. In a cohort Markov model, all Valuate expressions could be in Transition because they are only called once per cycle.

Using profiles to represent the subgroups

In the original formulation, the 8 subgroups (4 age groups for each of the 2 risk levels) were implemented by creating 8 copies of the 4 intervention Markov models. In DICE, the subgroups are implemented using profiles. A profile is a set of factors that characterize a particular type of person to be simulated. In this model each profile consists of an age group and a risk level. Because DICE executes the model for each profile, creating copies of the model is unnecessary. It does this by treating each profile characteristic as a condition (in this case, *AgeGrp* and *RiskStratum*) and setting their values to those of each profile in turn. These are used in Transition to find the probability of breast cancer (*pBreastCa*) without chemoprevention in input table *itblBreastCaProb* using INDEX (*itblBreastCaProb*, *AgeInterval*, *RiskStratum*). *AgeInterval* is used instead of *AgeGrp* because age will increase during the simulation, and this preserves the stratum-defining group. *Age* increases by 1 each cycle and *AgeInterval* is updated using the expression 1+(*Age*>54)+(*Age*>59)+(*Age*>64)+(*Age*>69)+(*Age*>74).

Verification and validation

A DICE model is validated in the same way other models are[11,12]: face validity is checked by subject matter experts, the implementation is carefully verified to detect and correct errors, and the forecasts made by the model are checked against the data used to build the model and, ideally, compared with external data. Given that the example used here had already been used in a NICE guideline, only verification was undertaken. The DICE model was verified by inspecting the DICE tables and using the tools supplied with the macro to ensure that all conditions and outputs were correctly named and all expressions met Excel syntax. Execution was verified by turning on the DICE logging utility and inspecting the text log that specifies every instruction executed together with

Table 4. Expressions for the Start event.

Type	Name	Expression
Condition	RRBreastCa	CHOOSE(IntervNum, 1, iRRTam, iRRRal, iRRAna)
Condition	CostInterv	CHOOSE(IntervNum, 0, iCostTam, iCostRal, iCostAna)
Condition	CostMonitor	IF(IntervNum>1, CostMonitor, 0)
Condition	pEndoCa	CHOOSE(IntervNum, pEndoCaNoTmt, pEndoCaTam, pEndoCaRal, pEndoCaAna)
Condition	pTEmb	CHOOSE(IntervNum, pTEmbNoTmt, pTEmbTam, pTEmbRal, pTEmbAna)
Condition	pFracture	CHOOSE(IntervNum, pFractureNoTmt, pFractureTam, pFractureRal, pFractureAna)
Event	Start	NEVER
Condition	NextEventTime	MIN(CurEventTime)
Condition	NextEvent	MATCH(NextEventTime, CurEventTime, 0)

its result. The DICE implementation was also presented to the team that built the original model at NICE.

Additional Elements

There is interest in most models in tracking the number of occurrences of certain items—review of these intermediate outcomes is often an important step in model understanding and validation. DICE keeps track of all changes to conditions and these can be output to a file. In this model, counts are kept of the endometrial cancers, thromboembolic events, fractures, breast cancers, and deaths. Each of these is stored in a "counter," a specialized condition that is reported along with other results. Counters are incremented by 1 whenever the corresponding item occurs. Additional counters can be added easily by inserting a row in the Outputs table and accruing that new output in Valuate or another appropriate event table.

Many of the inputs used in a model are subject to parameter uncertainty, and it is of interest to examine how much that uncertainty affects the results. One approach to analyzing this is to vary a parameter's value across its uncertainty range and assess the resulting variation in outputs in a deterministic sensitivity analysis (DSA). This is simple to complete in DICE because each input parameter's value is stored in a condition and can be varied to whatever other value is deemed appropriate. To address uncertainty across all parameters, a probabilistic sensitivity analysis (PSA) selects a value for each parameter from its range and reruns the analysis with that new set of values. To do this in DICE, the initial value of each parameter is replaced with a formula that implements the selection, and the number of replications is set to whatever the analyst wants. A special condition, *DoPSA*, is used as a flag that enables the PSA when it is set to 1.

Table 5. Main expressions in the valuate event.

Type	Name	Expression
Event	Valuate	Never
Condition	AliveHalfCycle	AVERAGE(PriorAlive, 1-Dead)
Output	caTmtCost	caTmtCost + AliveHalfCycle*CostInterv
Output	caCostMonitor	caCostMonitor + AliveHalfCycle*CostMonitor
Condition	BreastCaHalfCycle	AVERAGE(PriorBreastCa, HealthyBreastCa)
Output	caBreastCaCost	caBreastCaCost + BreastCaHalfCycle*CostBreastCa
Output	daEndoCaCost	daEndoCaCost + AliveHalfCycle* pEndoCa*CostEndoCa
Output	daTEmbCost	daTEmbCost + AliveHalfCycle* pTEmb*CostTEmb
Output	daFractureCost	daFractureCost + AliveHalfCycle* pFracture*CostFracture

Apart from parameter uncertainty, the many assumptions made in the design of a model generate structural uncertainty. One type of structural uncertainty arises from simplifying assumptions that can be parameterized. For example, in this model it was assumed that 50% of the women would stop adhering to treatment after 1 year, but it may be of interest to examine whether the timing of that treatment stop matters (the percent stopping is handled in the DSA and PSA). In DICE, the impact of this kind of assumption can be tested in a scenario analysis by creating a condition, say *NonAdhereTime*, and varying it to other plausible values. Because the *NonAdhere* event will be triggered at whatever time is set, no modifications of formulas are required. Indeed, a range of stopping times could be enabled by treating the initial *NonAdhereTime* as the first occurrence of *NonAdhere* and then at that event setting the next time of occurrence (and percent of who will stop then). Of note, these times need not coincide with the cycle time because time is treated continuously and more complex distributions using a stopping hazard (eg, a Weibull) are just as easily implemented. A scenario analysis can similarly be undertaken for any parameterizable structural assumptions of concern. This includes the cycle length, which not only could be changed to other durations but also could be made variable during the simulation so that it accords better with the timing of what is happening.

Structural uncertainty can also arise from design decisions that are difficult to parameterize. For example, when this model was specified, it was decided that the treatment side effects would only modify costs and not quality of life or mortality. To test the impact of these simplifying structural assumptions, events could be created for each of the side effects. At each cycle, each one is triggered and the proportion of the cohort experiencing a given one is processed, including whatever additional consequences may be considered appropriate.

Indeed, the decision to use a Markov cohort approach is itself a structural assumption, and it too can be easily tested using DICE. An alternative approach can be implemented within the same model. For example, a state-transition individual level model could be created by copying the Transition table, renaming it, say, to "Microsim" and modifying the expressions that apply the probabilities to test them against random numbers instead. With this individual implementation, greater heterogeneity than the 8

subgroups could be incorporated to check its effect and more complex patient pathways could be enabled. A simple toggle condition that sets Transition or Microsim to occur, one at "cycle" times and the other at "Never," can then be used to execute one or the other structure and compare their results using the same set of inputs and other assumptions. By the same token, the model could be designed as a discrete event simulation to explore whether the forced periodic timing of events and the inaccuracy it introduces significantly alters the results. This third structure can be started in an InitiateDES event and the toggle condition can be expanded to include it among the structural choices to make.

Results

Maintaining the Advantages of the Markov Model

The simplicity of the Markov approach is fully preserved in the DICE simulation. The states are represented as conditions and the transition probabilities are applied in a transition event. All the specifications are entered as text in simple tables, which are readily laid out in spreadsheets. The user interface remains a spreadsheet, ensuring that simplicity and comfortability remain.

Addressing the Disadvantages of the Markov Model

The DICE simulation simplifies the design and coding of the model. Because the DICE simulation does not limit the modeler to states and transitions among them, it is feasible to extend the Markov approach to a hybrid model that includes various events to represent occurrences such as stopping or switching treatment. Given that events are a core component of a DICE model, there is no barrier to treating occurrences as events. This facilitates the design of the structure, is considered easier to build, and yields a model that may be more transparent to reviewers than trying to incorporate events indirectly within states.

In contrast to the traditional implementation that uses rows as proxies for time, the DICE expressions are written only once because time is managed via looping in a short macro. This can help minimize error and facilitate verification and debugging. Because all components of an expression can be named (which is difficult to do when the formula is repeated many times with the reference cells changing at each repetition), a reviewer can "read" the model without having to extensively trace all the elements, sometimes across many worksheets.

DICE models are relatively quick to build because the entire model is specified in a few text tables. This makes DICE a useful tool for model validation via reimplementing the model, an area of increasing interest among modelers. This was demonstrated here in that the DICE model was able to reproduce the original Markov model.

Sensitivity analyses are completed without additional complexity. In addition, DICE supports extensive structural sensitivity analyses, even to the extent of carrying completely different structures in the same model. Modifying a DICE model is as simple as adding, deleting, or changing rows in a table—no relinking or other manipulation is needed. All of this can be done in standard MS Excel without requiring additional software. Of course, the DICE macro can also be implemented using other software such as Python, R or C#.

A more extensive redesign could address some of the simplifying assumptions made to facilitate the original implementation of the Markov model. For example, the AEs were not explicitly modeled—their only consequence was to add cost. In real life, of course, they would have health consequences, might alter treatment (eg, stop chemoprevention), and so on. These more realistic implications could be implemented in the DICE model by creating additional events to reflect each one, such as EndometrialCancer, Thromboembolism, or Fracture. In each event, whatever consequences were judged appropriate could be specified without abandoning the cohort Markov approach. The same could be done for the transition to breast cancer and for clinical events such as DEXA scans, monitoring, and mammography. Converting from a cohort approach to an individual level model would allow for better handling of heterogeneity in the characteristics that determine risk and mortality. This conversion can be achieved using DICE by simply changing how the transition probabilities are applied, while retaining the structure, events, and conditions. Individual patient profiles are automatically handled by the DICE macro.

Discussion and Recommendations

The cohort Markov approach to decision-analytic modeling has gained extensive popularity because it is relatively straightforward to specify and can be implemented in a spreadsheet without requiring specialty, and often expensive, software.[13] Given widespread familiarity with spreadsheets, this facilitates review and use of the models by others, particularly HTA agencies who must assess if the model predictions are reasonable and unbiased (in fact, we are not aware of any DICE model having been rejected by an HTA agency). These advantages are, however, tempered by the restrictions imposed by the Markov method itself and the nature of spreadsheet software that computes all formulas at once rather than in the sequential manner reflecting occurrences over time. The DICE simulation approach retains the advantages of spreadsheet modeling but also removes many of the Markov modeling constraints, as this article shows.

The use of spreadsheets for modeling can lead to slow execution speed when their calculations are limited to a stepped process as implemented through DICE. By default, spreadsheets trigger all calculations every time something changes. Because decision-analytic models involve many changes over time, the volume of calculations rapidly grows, especially when PSA is conducted. The DICE macro minimizes this by reading all model components into memory and turning off automatic worksheet calculation. The modeler can substantially help by minimizing the number of times the macro is forced to go back to a worksheet to read a value or carry out a calculation. A major step in this regard is to avoid referencing cells not included in the model tables and to ensure all required calculations are specified within the DICE tables. On a laptop running Office 365 32-bit with the current implementation of the DICE macro (version f), this model takes 7.6 seconds to run a replication with 8 profiles and 4 interventions. A PSA with 1000 replications takes 1.9 hours (original model runtime of 17 minutes). Of course, the execution can be massively sped up by rewriting the DICE macro in a compiled language (eg, a C# implementation exists but was not used in this exercise) and limiting the spreadsheet to the specification of the tables.

The need to define conditions and events in a manner that is conceptually meaningful, even if it remains a simplified representation of reality, can be challenging. Clearly this relies on the circumstances being modeled, but in most circumstances, we have found this can be undertaken, if not always with ease, then certainly with some forethought. A general limitation of the DICE approach (although not applicable to this exercise) is that it is not designed to implement full resource-constrained discrete event simulations or agent-based simulations. Both of these require simultaneous representation of many entities or agents and would be difficult to specify using DICE, at least in the

simple tabular form that provides one of its main advantages. These are, of course, limitations applicable to any Markov model.

Conclusion

Although there is an increasing use of individual simulation for HTA, the cohort Markov approach remains very popular despite its limitations. This exercise helped to demonstrate that a DICE simulation can provide what the original modelers considered a simpler, more transparent, more flexible modeling approach. Additionally, with DICE simulation it is possible to design and implement a model which best suits the problem at hand—from survival partition models to pure cohort Markov models to hybrids and full individual simulations. DICE simulation can provide capabilities that extend Markov models, improving the information provided for HTAs. It is hoped that this exercise helps to illuminate the advantages of DICE simulation and encourages the use of this free platform.

Article and Author Information

Accepted for Publication: March 4, 2020

Published Online: June 17, 2020

doi: https://doi.org/10.1016/j.jval.2020.03.009

Author Affiliations: London School of Economics, London, England, UK (Caro, Woods, McGuire); Evidera, Boston, MA, USA (Cairo); National Institute for Health and Care Excellence (NICE), London, England, UK (Maconachie, Naidoo).

Author Contributions: *Concept and design:* Caro, Woods, Naidoo, McGuire
Acquisition of data: Caro, Maconachie
Analysis and interpretation of data: Caro, Naidoo
Drafting of the manuscript: Caro, Maconachie, Woods, Naidoo, McGuire
Critical revision of the paper for important intellectual content: Caro, Maconachie, Woods, Naidoo, McGuire
Statistical analysis: Caro
Administrative, technical, or logistic support: Woods
Supervision: Naidoo

Conflict of Interest Disclosures: Dr Caro receives a salary from Evidera, a consultancy that provides services to industry. Dr Caro reported receiving grants from Horizons 2020 during the conduct of the study. Mr Maconachie and Dr Naidoo are employed by the National Institute for Health and Care Excellence (NICE), an HTA and guideline writing body for the publicly funded health and social care system in the United Kingdom. No other disclosures were reported.

Funding/Support: This project was supported by grant 779312 from the European Union's Horizon 2020 research and innovation programme. The views expressed in this article are those of the authors and not necessarily those of the European Commission or NICE (www.nice.org.uk).

Role of the Funder/Sponsor: The funder had no role in the design and conduct of the study; collection, management, analysis, and interpretation of the data; preparation, review, or approval of the manuscript; and decision to submit the manuscript for publication.

REFERENCES

1. Siebert U, Alagoz O, Bayoumi AM, et al. State-transition modeling: a report of the ISPOR-SMDM modeling good research practices task force-3. *Value Health*. 2012;15:812–820.
2. Sonnenberg FA, Beck JR. Markov models in medical decision making: a practical guide. *Med Decis Making*. 1993;13:322–338.
3. Caro JJ, Moller J. Advantages and disadvantages of discrete-event simulation for health economic analyses. *Expert Rev Pharmacoecon Outcomes Res*. 2016;16:327–339.
4. Caro JJ. Discretely integrated condition event (DICE) simulation for pharmacoeconomics. *Pharmacoeconomics*. 2016;34:665–672.
5. Woods B, Sideris E, Palmer S, Latimer N, Soares M. NICE DSU technical support document 19. Partitioned survival analysis for decision modelling in Health Care: A Critical Review. http://nicedsu.org.uk/technical-support-documents/partitioned-survival-analysis-tsd/. Accessed January 4, 2020.
6. Caro JJ. Pharmacoeconomic analyses using discrete event simulation. *Pharmacoeconomics*. 2005;23:323–332.
7. Möller J. Cooking up a transparent model following a DICE recipe. *Pharmacoeconomics*. 2019;37:1341–1347.
8. Moller JM, Davis S, Stevenson M, Caro JJ. Validation of a DICE simulation against a discrete event simulation implemented entirely in code. *Pharmacoeconomics*. 2017;35:1103–1109.
9. National Institute for Health and Care Excellence. Addendum to Clinical Guideline 164, Familial breast cancer: Methods, evidence and recommendations. Appendix N Economic modelling report. https://www.ncbi.nlm.nih.gov/books/NBK550790/. Accessed January 3, 2020.
10. Caro JJ, Möller J. Adding events to a Markov model using DICE simulation. *Med Decis Making*. 2018;38:235–245.
11. Eddy DM, Hollingworth W, Caro JJ, et al. Model transparency and validation: a report of the ISPOR-SMDM modeling good research practices task force-7. *Value Health*. 2012;15:843–850.
12. Caro JJ, Eddy DM, Kan H, et al. Questionnaire to assess relevance and credibility of modeling studies for informing health care decision making: an ISPOR-AMCP-NPC good practice task force report. *Value Health*. 2014;17:174–182.
13. Caro JJ, Briggs AH, Siebert U, et al. Modeling good research practices—overview: a report of the ISPOR-SMDM modeling good research practices task force-1. *Value Health*. 2012;15:796–803.