Aggregation of Monte Carlo Models to Support Meta-Modelling

Dr Rhys Williams and Cressida Johnson
Policy and Capability Studies, Dstl Farnborough, Hampshire, GU14 OLX

Stochastic Processes

Many combat models use stochastic processes to account for combat uncertainties, such as:

- enemy actions
- equipment imperfections
- weather conditions

Monte Carlo modelling is one way of incorporating stochastic processes - the analyst supplies a probability distribution that represents the likelihood of an event occurring and the algorithm randomly samples against it to determine whether it actually occurs. Monte Carlo models are run many times, producing output in the form of distributions, which represent results corresponding to different combinations of uncertain events. The need for multiple runs is an inconvenience - this study was designed to explore methods of mitigating this. Output distributions from high fidelity models, such as an engagement outcome, are often used as input for the stochastic elements of more aggregated, lower fidelity models. This data transfer process is illustrated in Figure 1.

Meta-models are simpler models that are designed to replicate or approximate particular processes within a more complicated model. A meta-model is thus a 'model of a model'. Meta-models' fast-running features facilitate the data transfer process.

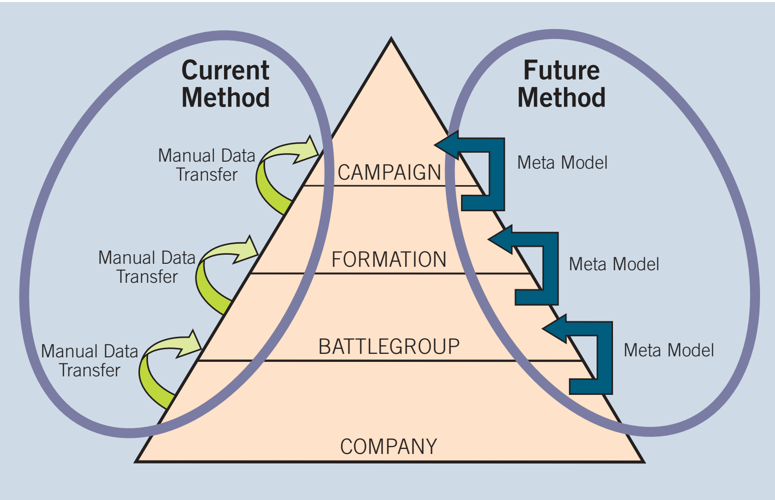

Figure 1. The model pyramid - each tier corresponds to a different level of aggregation

Aggregation Methods

Aggregation methods can simplify complicated sequences of stochastic processes by merging them - combining probability distributions to produce output distributions directly, without sampling. Aggregation methods have wide potential. The aim of this study is to investigate the application of aggregation techniques to stochastic models by characterising stochastic processes in ways that make them amenable to particular aggregation methods.

Broadly, the stochastic processes that we considered can be classed as:

- **Non-branching** - producing outputs that remain fixed for the entire run e.g. weather conditions
- **Branching** - producing outputs that impact on other processes embedded in the model run

We considered the following aggregation methods for each class of stochastic process:

- **Maximum/minimum aggregation** to determine the probability of the maximum or minimum of more than one uncertain event occurring
- **Additive Aggregation** to determine the probability of the sum of two uncertain quantities occurring
- **Sequential/conditional aggregation** to determine the probability of several dependent events occurring

Aggregation methods facilitate meta-modelling by focusing on particular stochastic processes, producing output that can be transferred between models.

Aggregation Methods - Non Branching Examples

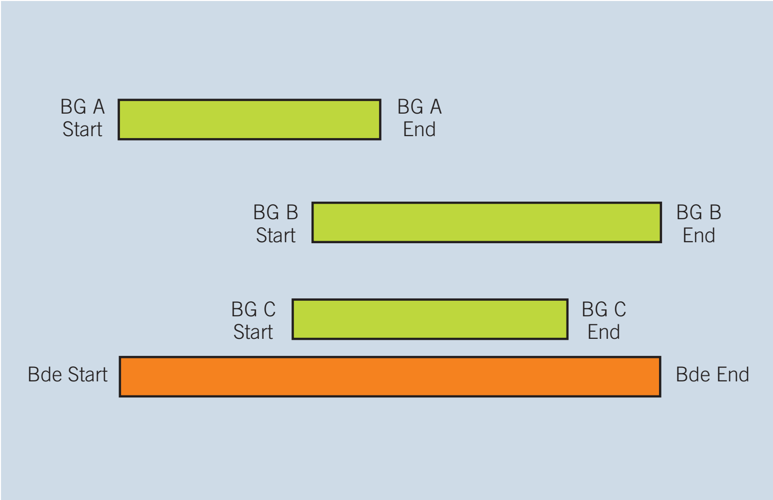

Figure 2. Maximum-minimum Aggregation

The task bars in Figure 2 are illustrative of issues that arise in event-stepped models. The bars show start and end times for tasks performed by battlegroups A, B and C. The lower bar illustrates the use of maximum/minimum aggregation methods to show the time taken by the higher-level brigade.

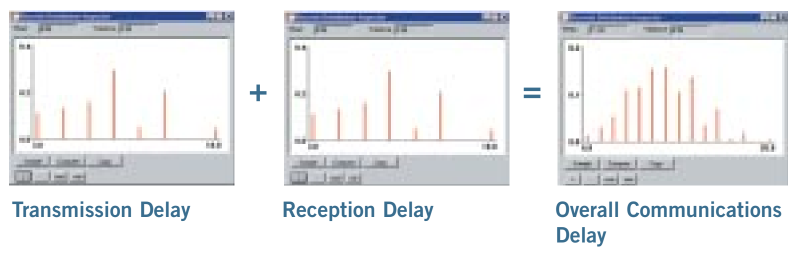

Figure 3. Additive Aggregation

Figure 3 illustrates the use of additive aggregation as a means of computing the overall delay, given distributions for its component (transmission and reception) delays.

Aggregation Methods - Branching Examples

Most stochastic processes result in some form of probabilistic branching. Treating these branches as conditional probability problems allows us to 'aggregate' our way down each branch, thus determining an output distribution for the end states.

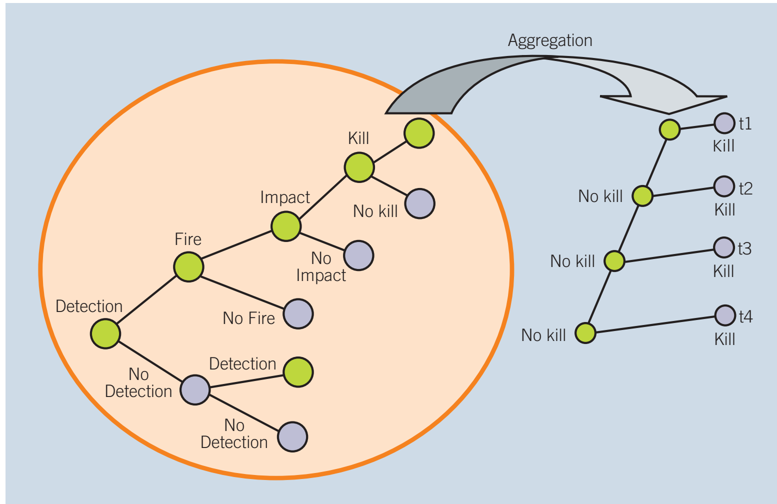

Figure 4. Sequential or conditional probabilistic branching

Figure 4 illustrates a multi-step process representing one-on-one engagement. Statistical computation allows us to aggregate the circled branching into the simpler representation on the right.

If the kill probability is known as a constant value, we can compute the kill probability at the nth time-step using a simple Markov computation. However, input distributions are not usually constant, but depend on variables such as time or distance between units. A more complex branching process is illustrated by Figure 5.

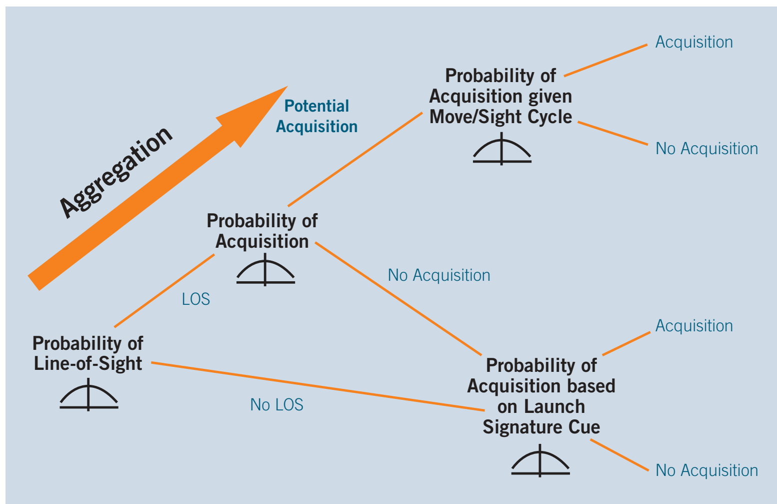

Figure 5. The Simbat Target Acquisition Algorithm

Figure 5 shows a more complicated detection algorithm, used by the Simple Battlegroup (SIMBAT) combat model. At each time-step, each unit attempts to acquire potential sightings by working through the steps shown. The probability distribution determining the likelihood of going down each branch is user-specified. We can compute the probabilities of reaching each end-state using conditional probability theory to aggregate the branches.

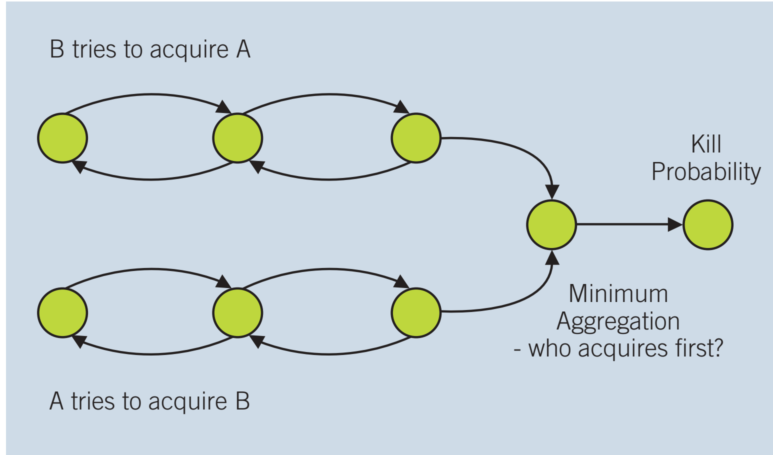

Figure 6. Simultaneous attempted target acquisition

Figure 6 conceptualises simultaneous detection attempts made by units A and B. The conditional probability computations are followed by a minimum aggregation method to compare detection times and determine which unit acquires the other first.

More Complex Interactions between Stochastic Processes

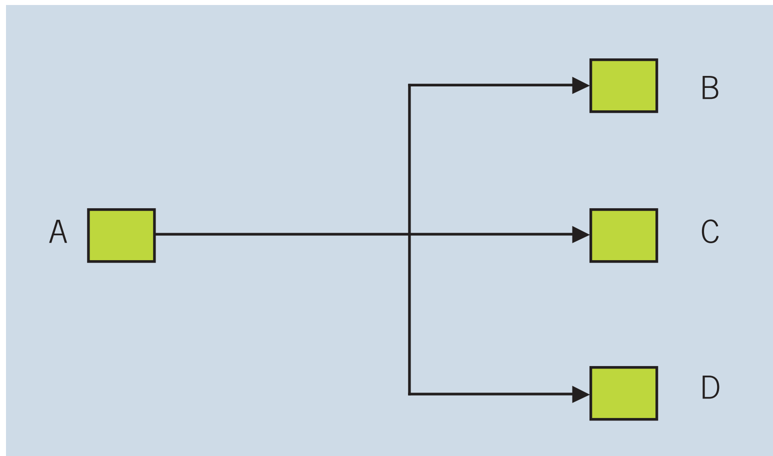

Figure 7. Process A could potentially lead to process B,C or D

The inter-relations between stochastic processes in complex combat models are difficult to visualise, as they may vary from one run to the next. One major issue to be dealt with by this study team, shown in Figure 7, is how to handle the fact that a given stochastic process may lead into different processes with each model run.

Conclusions

This study developed a set of generic concepts for visualising stochastic processes that are used in combat modelling. Those representations were composed to be amenable to standard stochastic aggregation methods. Our follow on task, from this initial scoping phase, is to test the utility of aggregation methods by comparing aggregated results with results obtained via Monte Carlo sampling.