

Presenting Uncertain Scenarios

by Bob Eberlein and Billy Schoenberg
isee systems, inc. Lebanon MA, USA

Abstract

When thinking about different scenarios and the implications of decisions made today on possible future outcomes, we often present scenarios composed of a single run - one value for each variable for each point in time. When paired with the appropriate discussion, this approach can be very effective, but it runs the risk of stakeholders focusing on specific numbers, and thinking our models are far more accurate than they are. That misperception of a model's utility can detract from a more systemic understanding of issues. By presenting scenarios with an uncertainty-based spread of outcomes, we make the discussion richer and are more likely to get to deep system insights. This paper outlines a relatively straightforward process for representing uncertain scenarios and shows an example of the type of interfaces that can be developed using these techniques.

Background

Sophisticated forecasting and projection techniques abound in today's business world. But even while Monte Carlo simulation and possible outcomes distributions are frequently used, what is used by senior management in making important strategic decisions is generally single-run based projections (see for example, En-Roads (Siegal et al 2023), Rethink Health (Milstein 2023) and most of the other System Dynamics Society Applications Award winning decision support interfaces). Whether the results of a simulation run are presented as median outcomes or (as is more commonly done) simply thought of as forward projections, they are overly narrow. Decision makers want a rich set of information and intuition to inform their actions, and going beyond a single value at a point in time will give good quantitative work more weight.

What we set out to show in this paper, is a way to compute and present ensembles of runs that represent the uncertainty inherent in the projections made given a single set of decisions. To make this useful, we need to compare these outcome spreads across different sets of decisions or expectations on business conditions. To do this we faced challenges related to computation and presentation. With some care, the computational challenges are easily overcome, and we will provide some detail on how we did this which

should be useful to others trying to do the same thing. The presentation challenges, for us, required small additions to the existing graphing toolset in Stella, though similar things can be done with other graphing toolsets.

Finally, to bring it all together, we show how our solution is implemented in a usable interface and can be utilized in discussion with stakeholders to explore decisions and their consequences. We will base this paper on an extremely simplified model that represents the outcomes of choosing different levels of capital investment and research and development activity. The engagement that inspired this work had many more dimensions, including target market areas, several R&D focus areas, and supply chain considerations. The fundamental ideas discussed apply without change in that context, there are just more choices and sources of uncertainty.

Multiple runs in one

In order to run a sensitivity in a single simulation, we arrayed the model by run. This approach is something the authors have long used, but never written or read about. Thus, though straightforward, it seems useful to say just a few words about it.

Start with the simplest possible model – a constant – and run sensitivity on that assuming a uniform distribution. The result of that sensitivity analysis is a number of runs, where the constant takes on a different value in each of them. Contrast that with a model that has a single one-dimensional array, where each array element is specified using a uniform distribution. If you look at the single variable across all runs of the sensitivity analysis, or the arrayed variable for a single run, each set of numbers will be uniformly distributed. In a word they are not distinguishable from one another.

This simple approach generalizes to any level of model complexity. Start with the sensitivity parameters, array them by the Run array (or add it to them if they are already arrayed), then follow their uses to ensure that everywhere they are used by other variables in the model are also arrayed by Run. Consider the very simple example shown in Figure 1. In this example *carrying capacity* is being varied as part of sensitivity. In the arrayed version all variables except the two additional constants *birth rate* and *average lifetime* are arrayed by Run. This is quite typical.

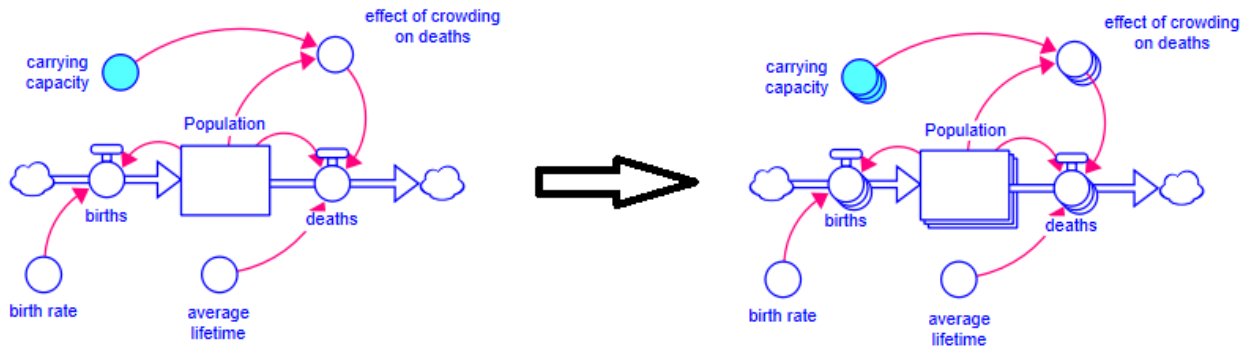


Figure 1. Making variables arrayed in order to run sensitivity in a single run

The values used for the arrayed version could be generated using a purely random Monte Carlo generator, but they can also be specified by determining the sensitivity values for the scalar version of *carrying capacity*. This allows the use of Sobol Sequences (Sobol, 1967, Press et al., 1992) which improves the exploration of the uncertainty space. Once the model is arrayed in this way, and the appropriate set of values for the uncertain parameters have been set, each time the model is run, the equivalent of a sensitivity simulation occurs. The elements of the array Run represent the individual runs that would have occurred in the original model.

The process for generating and importing these value sequences will vary depending on the software package being used. With Stella, we create an abstract model with a vector of uniform draws on $[0, 1]$ where the length of that vector is equal to the number of uncertain parameters we have in our model, and the number of draws is equal to the length of the array Run in the main model. We use $[0, 1]$ because we will scale the value when it used in the model, and a uniform distribution to maximize variety. Our approach places no restrictions on the type of distribution used, though scaling would need to be done differently with non-uniform distributions. Using Sobol sequences exactly matching the number of parameters and runs is important, as these sequences are designed based on these two numbers. We run a sensitivity on our abstract model (SobolInit.stmx), and then copy the sensitivity parameters into an import file for the main model (CapExAndR&DArrayed.stmx). The models that do this are included as part of supplementary material.

The display of outputs, which we will discuss in the next section, is facilitated by the use of a PERCENTILE function which delivers specified percentile cutoff points. For example,

$$\text{PERCENTILE}(\text{Revenue}[*], 25) \tag{1}$$

will return the 25th percentile (the value of revenue for which 25% of the runs are below), while

$$\text{PERCENTILE}(\text{Revenue}[*], 75, 25) \quad (2)$$

will return the difference between the 25th and the 75th percentile. With these computations in place, we can display the spread over the distribution, though as you will see in the example we typically use the minimum, 25th percentile, median, 75th percentile and maximum.

To facilitate the easy creation of a standard set of percentiles, we typically define a new numeric Array PercentileCutoffs. That dimension is given a number of elements which corresponds to the number of percentile boundaries we want to present. This then allows us to create output variables arrayed by PercentileCutoffs which we use in the data visualization as described below. The basic equation for the displayed variable is that shown in Equation 3, where Revenue is arrayed by Run as described above.

$$\begin{aligned} \text{revenue_percentiles} [\text{PercentileCutoff}] = \\ \text{PERCENTILE}(\text{Revenue}[*], \\ \text{percentile_mapping}[\text{PercentileCutoff}], \\ \text{percentile_mapping}[\text{PercentileCutoff}-1]) \end{aligned} \quad (3)$$

A model using these formulations is included in the supplementary material (CapExAndR&D.stmx).

Alternatively, one can formulate the percentile cutoffs using macros as shown in Equation 4¹. This has the advantage of not requiring the input percentile_mapping for every percentile computation. The macro itself is very simple as is shown in equation 5. This alternative formulation is also included in the sample model.

$$\begin{aligned} \text{macro_revenue_percentiles} [\text{PercentileCutoff}] = \\ \text{PERCENTILE}(\text{Production.revenue}[*], \\ \text{macro_percentile_mapping}(\text{PercentileCutoffs}), \\ \text{macro_percentile_mapping}(\text{PercentileCutoffs} - 1)) \end{aligned} \quad (4)$$

$$\begin{aligned} \text{out} = \text{IF in} \leq 1 \text{ THEN } 0 \\ \text{ELSE IF in} \leq 2 \text{ THEN } 25 \\ \text{ELSE IF in} \leq 3 \text{ THEN } 50 \\ \text{ELSE IF in} \leq 4 \text{ THEN } 75 \\ \text{ELSE } 100 \end{aligned} \quad (5)$$

¹ There is a simpler solution still to doing this that would require some backend work updating the graphing libraries. That would allow us to remove the percentile computations from the model completely.

Displaying outcome ranges

There are three common graph types for displaying outcome ranges: confidence (percentile) graphs, histograms, and box and whisker plots, as shown in Figure 1.

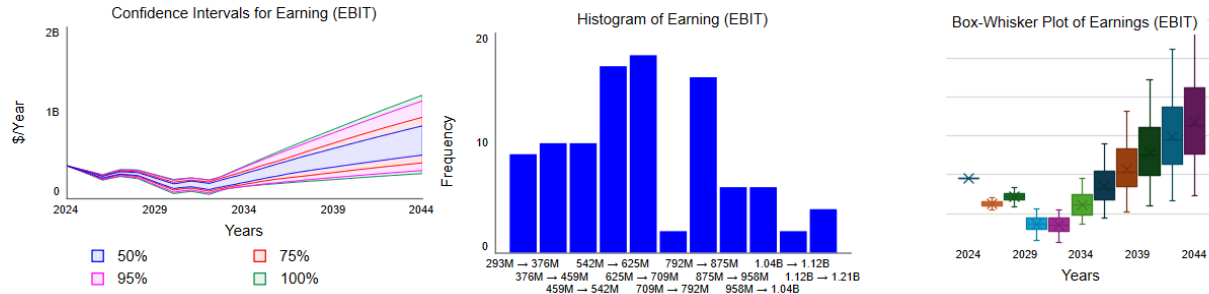


Figure 2. Common graph types for showing uncertainty.

Histograms provide the most detailed information, but only at a single point in time. Making comparative runs with histograms, while possible, is quite confusing as the bar counts interleave one another and make it hard to see the overall shape of the distribution of outcomes in each scenario.

Confidence graphs provide information at all points in time continuously, but usually take up the entirety of the graph's internal space in doing so. This makes side-by-side comparison very difficult (the only way to really do this is show two graphs next to, or on top of, one another).

Box and Whisker plots only do distinct points in time but do so in a relatively compact format. This makes it possible to compare outcome ranges for different scenarios, and is the direction we chose, though with a slightly different format.

In the work that inspired this paper, we used very few tabular presentations of output, and when we did, we tended to display only median results. Our focus was always strongly on outcome patterns, not specific numbers, though we did often end up reading from graphs (by scrubbing) during discussion.

To create something with the characteristics of a Box and Whisker Plot, we made use of what we call comparative stacked bar graphs. In these each bar displays the incremental percentile from Equation 2 for 0, 25, 50, 75, 100 as shown in Figure 3. In this graph we used dark red to display the minimum, red the median, and grays for outcomes above the median. Colors are a fairly subjective choice, but the contrast below and above the median makes it easier for people to distinguish the upside. The consistency below the median takes some of the emphasis off the downside, without hiding it altogether.

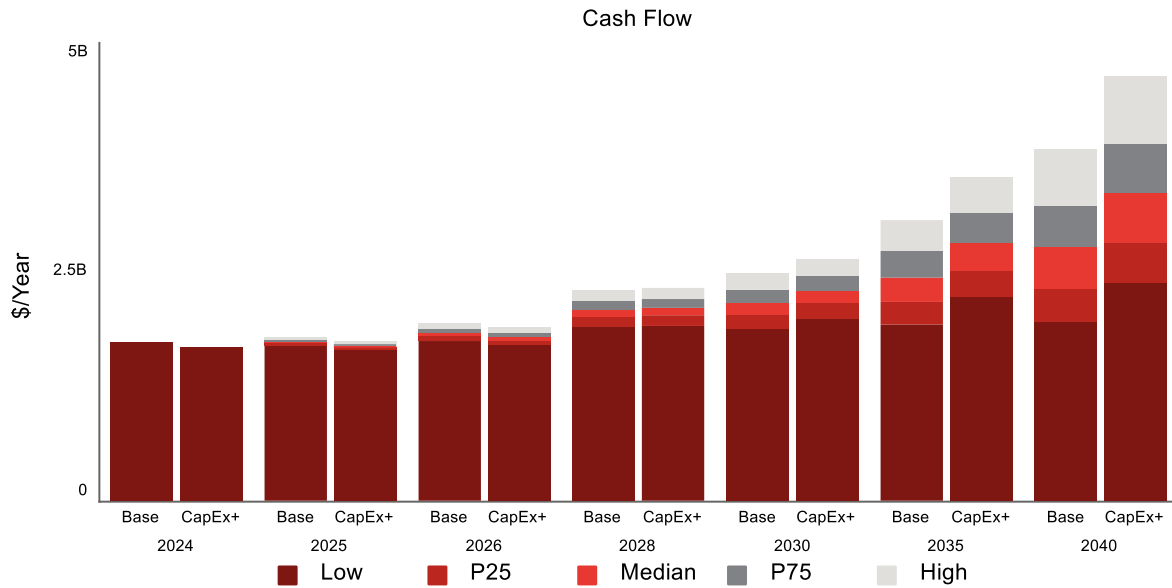


Figure 3. Using comparative stacked bar graphs to display outcomes.

Selecting uncertainty bounds

By choosing different uncertainty bounds, we can create and explore scenarios interactively with stakeholders. Because of the way that we generated the parameter sets for each run, this can be done by setting a minimum and maximum range on the uncertain parameter. We then use the formula:

$$\text{Uncertain_parameter}[\text{Run}] = \text{min_value} + (\text{max_value} - \text{min_value}) * \text{parm_draw}[\text{Run}] \quad (6)$$

where $\text{parm_draw}[\text{Run}]$ is the value of the parameter for the current run created uniformly on $[0, 1]$ according to the steps described in the section Multiple runs at once. With this in place, we can present the user with a simple interface such as that shown in Figure 4. Note the use of qualitative labels on the uncertainty ranges. This can help engender an attitude of exploration in clients, which helps in understanding the nature of long-term strategic tradeoffs.

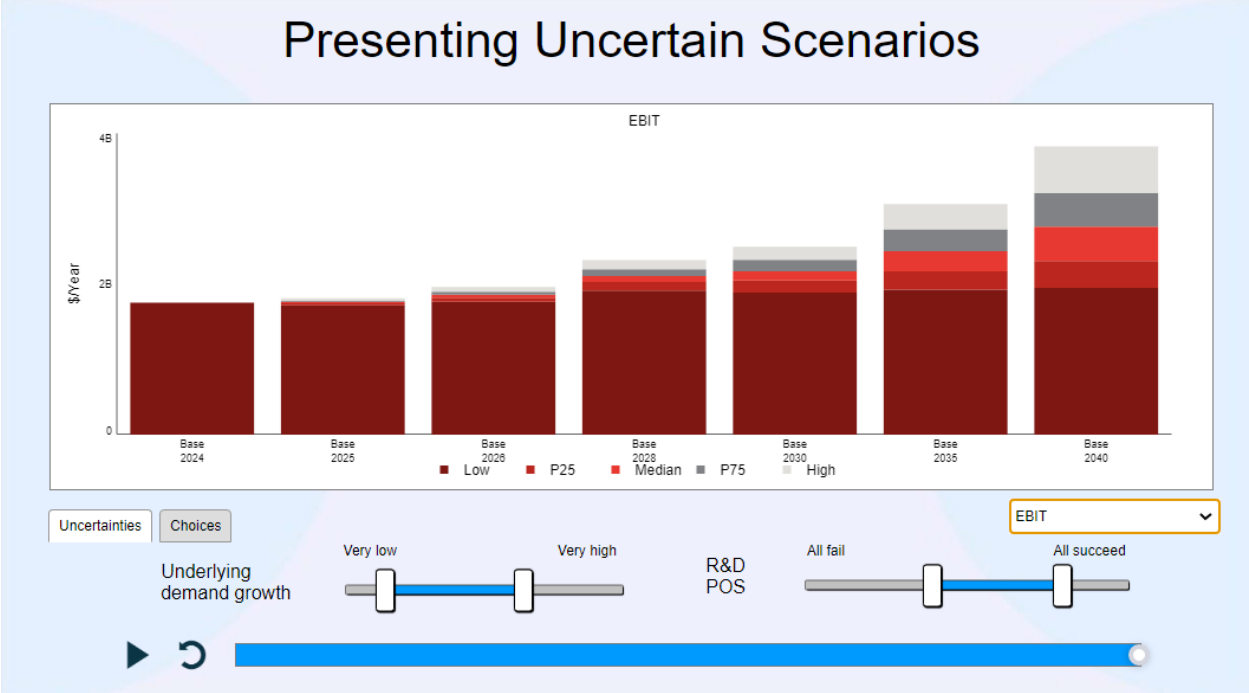


Figure 4. Using comparative stacked bar graphs to display outcomes.

Example

For the example in the supplementary material, we use a simplified model showing the relationships between investments and financial outcomes. The model is a vastly simplified version of one developed as part of a consulting engagement. The overall architecture or the model has 4 modules as shown in Figure 5.

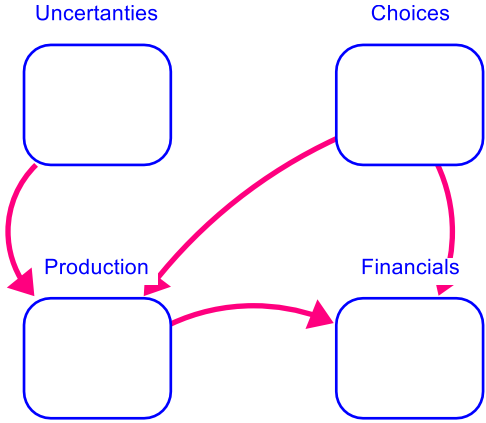


Figure 5. High level views of example model.

The Production module captures both R&D and capital investment and the effect they have on revenue as shown in Figure 6.

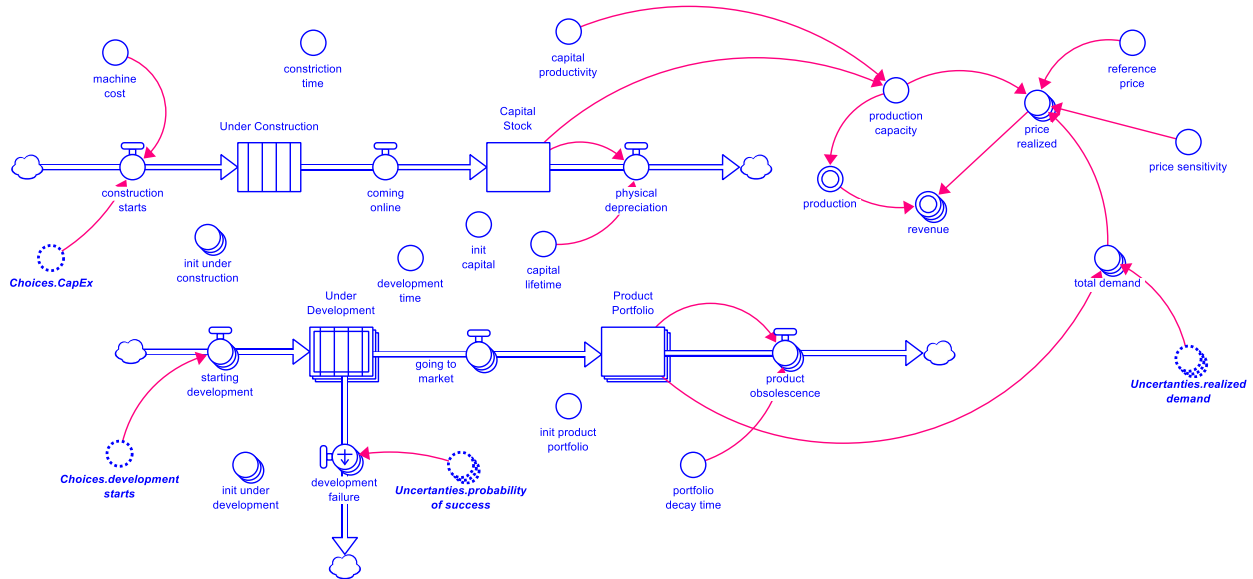


Figure 6. Production and revenue resulting from investment.

In this model the R&D portfolio and products in the market is arrayed by Run to incorporate the uncertainty described in this paper. The capital stock and productive capacity as the same for each run. Additional uncertainty could certainly be introduced there, but we wanted to keep this to a minimum to make it easier to understand the model.

An interface was created for the model and is available at:

<https://exchange.iseesystems.com/public/bob1/uncertain-scenarios>

A screenshot of the interface is shown in Figure 4. It is compact and relatively easy to use, but it is also somewhat under constrained in terms of how much investment is available in any given year. Still, it can be used to make the point that R&D investments tend to take longer to give results and broaden the spread of possible outcomes more than they move the median outcome. A statement that is fairly obvious to those working in System Dynamics, but one that is not necessarily easy to convey using simple line graphs.

One of the things that is noticeable when using this interface is that as each run is made the bars fill in one at a time for each year as the simulation progresses. For this sample model this progression is apparent only because we set the model to run slowly. For larger models, however, the speed can be a result of model size, and the fact that each run is actually a set of parallel runs run together. This type of animation can be contrasted with a regular sensitivity run, in which the sensitivity graph fills in all times at once or redraws itself for all times as more runs become available. That type of animation is much less satisfying when working with clients.

Conclusions

In this paper we have shown a technique that, with a small extension of the standard System Dynamics toolset, improves conversations around choices we can make today and their impact on future outcomes. By presenting a range of outcomes, rather than a single point value, discussions can go beyond simple target comparisons to the time profile of results. In a sense, this gets us to the intent for most work in System Dynamics – understanding how our actions and the structure of the system result in observed outcomes. The conversation is better, however, because stakeholders are not as strongly anchored on specific realized outcomes.

References

Milstein, Bobby (2023) "What Does It Take to Be Good Stewards of the Systems That Shape Our Lives Together?" Proceedings of the 2023 Conference of the System Dynamics Society, <https://proceedings.systemdynamics.org/2023/supp/S1044.pdf>

Press, W.H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992) "Numerical Recipes in C: The Art of Scientific Computing, 2nd ed." *Cambridge University Press, Cambridge, U.K.*

Siegel, Lori, Chris Campbell, Adem Delibas, Sibel Eker, Tom Fiddaman, Travis Franck, Jack Homer, Andrew P. Jones, Charles Jones, Joshua Loughman, Stephanie McCauley, Elizabeth Sawin, Chris Soderquist, John Sterman (2023) *En-ROADS Technical Reference* <https://docs.climateinteractive.org/projects/en-roads-reference-guide> (Accessed March 15, 2024).

Sobol', I. Y. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4), 784-802.