Developing Quantitative Scenarios

A Handbook for Accidental Practitioners

Stockholm Environment Institute
May 2007

SEI STOCKHOLM ENVIRONMENT INSTITUTE

For more information about this document, contact Eric Kemp-Benedict at erickb@sei-us.org.

Table of Contents

Introduction

This handbook introduces some techniques in use at the Stockholm Environment Institute (SEI) for planning and carrying out a scenario quantification exercise. The approach is practical, and is intended to assist diverse groups carrying out their own quantitative scenario exercises. It is addressed to "accidental practitioners," people who are not professional modelers or scenario developers, but who are being called upon to develop quantitative scenarios in the course of their work.

The focus in this handbook is on an approach to scenario model development that SEI employs in its own work and continually refines over time. The goal in this approach is rapid, custom model development targeted to the needs of model users in the context of a project with time and budget limitations. As these are conditions that apply in many scenario exercises, these techniques should be widely applicable.

The main strategy for achieving rapid, targeted model development is to use indicators as a way to focus the development. In this approach, the goal of the model development is seen as providing needed indicators to the people who will eventually use the model. By focusing on indicators the task of model development becomes that of creating a parsimonious model that calculates the outputs required by the model users – the indicators – in a technically defensible way.

Broadly, the steps involved in developing quantitative scenarios as described in this handbook are as follows, with a range of likely times specified in parentheses.

1. Specify the boundaries of the study (½ day – 1 day),
2. Select indicators (½ day – 1 ½ days),
3. Decide on a model structure (½ day – 2 days),
4. Estimate the time, decide on a schedule, and revise the scope if necessary (½ day – 2 days),
5. Iteratively develop, test, document, and release the sub-models (variable),
6. Release the final set of quantitative scenarios (variable).

The entire activity is likely to take more than a month and, depending on the level of sophistication of the final model, can take years. The focus in this handbook is on exercises that take at most a few months.

Getting Started

A few relatively quick tasks should be completed up front before thinking about the approach to quantifying the scenarios.

Before starting

Before starting a quantitative scenario exercise, some tasks must already be accomplished:

- Identification of the audience,
- Selection of the time horizon,
- Identification of major themes,
- Creation of a scenario framework,
- Initial discussion of indicators.

These tasks set the stage for the scenario development, and help to bound the quantitative analysis. A further task is:

- Survey of available material.

It may be that the survey will show that the necessary quantitative scenarios already exist, so that no additional work needs to be carried out. More typical is the situation in which some material can be adapted from existing studies, either to inform or simplify the modeling exercise.

Specifying the boundaries

With the prerequisites in place, the first relatively quick task is to specify the boundaries of the quantification, including:

- Time boundaries (base year, final year, and intermediate years),
- Geographic boundaries (for example, regions, countries, watersheds),
- Thematic boundaries (themes explored in the scenarios that require quantitative elaboration),
- Range of futures (for example, a narrow set of scenarios exploring minor variations around a particular policy, vs. a wide set of scenarios exploring

fundamentally different visions of the future); this should include the number of scenarios and their characteristics,

- Audiences and intended use (for example, for decision support or for communicating a positive vision of the future).

These should be readily available from the work that precedes the quantification. It is a good idea to collect it in one place at the start of the quantitative scenario development, and make sure that there is agreement on the boundaries of the exercise among those participating in the scenario development and using the scenario outputs.

Collecting indicators

The approach to scenario quantification described in this handbook uses indicators as an organizing framework for model development. Before planning the scenarios in earnest, it is necessary to have at least a preliminary indicator set collected from the people who will be using the scenario outputs. Ideally, there should be a qualitative indication of how the indicators are expected to behave in each of the scenarios.

For the purposes of this handbook, an indicator can be defined as "anything you want to see on a graph."[1] Slightly more concretely, it is a representation of the state of the system being studied that helps to distinguish between different scenarios. The indicator set should be chosen so that it supports the purpose of the scenarios, for example, as decision support, or to indicate the severity of a future challenge or risk.

It is likely that the initial list of indicators will be unreasonably long, and that some indicators cannot be quantified. This is not a problem at the initial stage, since planning for the quantification will include selecting a subset of indicators from the full list, as described in the next section.

1 This definition was helpfully provided by a workshop participant. Indicators are a topic of study all their own. However, most of the debates about indicators have to do with the proper way to create and use composite indicators, which aim to show the state of a complex system in a single number. In this handbook the focus is on collections of indicators, which are easier to get agreement on than composite indicators.

Selecting Indicators

The approach described in this handbook to planning quantitative scenarios uses indicators to build an organizing framework.

Why use indicators?

The scenario modeler's job is to create a model that reasonably represents the system under study. However, only other modelers and some area experts will be interested in the details of how that is carried out. What most users of the model are interested in are the indicators, that is, numbers that summarize important features about the system. Organizing the modeling effort around the indicators focuses the project on the output of most interest to the users of the scenarios, while also providing a way to track overall model development.

Selecting and prioritizing

The initial list of indicators is likely to be longer than is practical, and is also likely to include some indicators that cannot be quantified, perhaps because they are inherently qualitative, or because the data are not available, or because they cannot be modeled in a technically defensible way. A practical way to select indicators is to use a two-step process, first selecting quantifiable indicators, and second prioritizing the remaining indicators.

The first step is to move all non-quantifiable indicators into a separate list. These indicators will not be outputs of the quantitative models. However, they are likely to play a role in the scenario narratives, and may also inform the development of the quantitative models.

The second step is to prioritize the remaining indicators. There is more than one way do this. One approach that has been used for project planning in other contexts is to use the so-called "MoSCoW Rules."[2] The acronym "MoSCoW" is a mnemonic for a set of rules:

- **M**ust include
- **S**hould include
- **C**ould include
- **W**on't include

2 It is a technique from the Dynamic Systems Development Method (DSDM) for software development.

The procedure is simply to take the list of indicators and put an M, S, C, or W next to the indicator. Those indicators with a W can be dropped from the active list of indicators, and those with an M placed at the top, for future planning.

An alternative to the four-category MoSCoW approach is to make a simple "keep/discard" decision. However, at this stage it is difficult to decide which indicators to definitely discard, since time estimates have not yet been constructed and it is unknown how much work can be accomplished.

Choosing a Model Structure

The high-level structure of the model must be decided before a schedule can be estimated. There are many ways to structure a scenario model, and each has advantages and disadvantages. Some model structures (not an exhaustive list) are shown schematically in Illustration 1.

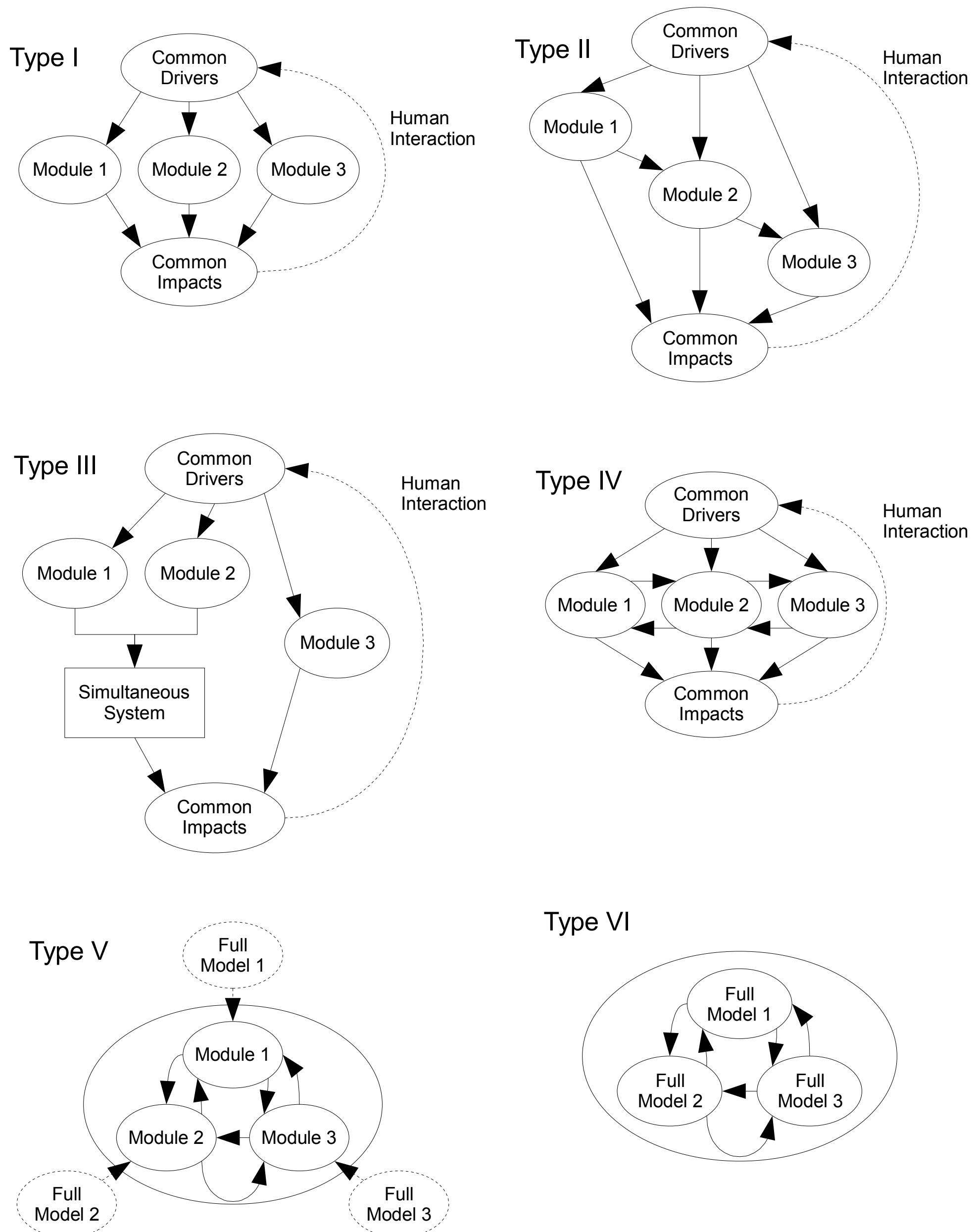

Illustration 1: Schematic model structures

Choosing a model structure

The list of model structures in Illustration 1 generally proceed from less complex and easier to implement to more complex and harder to implement. Model type VI is closest to the structure of an integrated assessment model. It endogenizes as much as possible and seeks to represent internally as many interactions in the external world as possible. With such models, interaction with the model user is usually via parameters that determine the strength of interactions. Model type V is a variation of model type VI in which reduced forms of full models are integrated together.

Model types I-IV, while not as sophisticated as model types V and VI, have much to recommend them when implementing scenario models. For many scenario studies, a structure similar to one of types I-IV is sufficient, and in some cases preferable, to the more complex structures shown as types V and VI. Model types I-IV are in general more flexible and adaptable than are model types V and VI,[3] which permits greater interaction between modelers and area experts while the model is being developed; can result in models that are better targeted to the needs of a particular study; and allows the implementation of strongly divergent visions of the future. While in some situations the inclusion of human interaction as a part of the modeling process is a drawback, in others – for example in a stakeholder process – it can be a necessity.

A further advantage of model type I is that it is possible for several people to develop the model in parallel. While there must be initial agreement on the types of inputs and outputs, once that is determined the modeling effort on the different modules can be carried out independently and simultaneously.

Types of models

Within the overall structure of a model there are many further choices for implementation. It is not possible to discuss all of the possibilities in a short handbook like this one. A short list of some of the more commonly-encountered model types is presented in this section, grouped under the headings "balances," "self-consistent frameworks," and "components."

3 An interesting alternative is the IFs model, which has a structure like that for type V and VI models, but which allows entire sub-models to be turned off and on. This provides much more flexibility than do other models of type V or VI. IFs is available from http://www.ifs.du.edu/.

Balances

The most basic feature of most models is a set of balances – for example, an energy balance – that expresses the conservation of some quantity. These are particularly important in sustainability scenario models because central to the sustainability question is whether use of limited resources is consistent with meeting long-term needs. Important balances include

- Matter balances (land, biogeochemical, water, waste),
- Energy balances (energy extraction, conversion, and demand),
- Trade, consumption, production, and other economic balances,
- Demographic balances.

Self-Consistent Frameworks

As shown in Illustration 1, many models are embedded within a framework that ensures self-consistency. For example, the framework may enforce the balances, or internalize behavioral rules. Some of the more common approaches to ensuring self-consistency are

- Matrix models (including economic input-output models and social accounting matrices),
- Linear and nonlinear programs, including goal programmming,
- Systems of differential equations (including system dynamics, hydrological, and material transport models),
- Computable general equilibrium (CGE) models (such as a global or regional trade model).

Components

Most models are built out of components, each of which might use a different modeling approach. Some of the more common varieties include

- Empirical regression models (including econometric models),
- Theoretical models (such as a model of pollutant transport),
- Probabilistic models (including Monte Carlo and Bayesian models),
- "Fermi Problem"/IPAT calculations.[4]

4 The physicist Enrico Fermi was known for giving seemingly impossible problems to his students (such as determining the number of piano tuners in the U.S. city of Chicago) that could be answered by breaking the problem down into a product of factors. These became known as "Fermi Problems." The IPAT formula is $I=P\times A\times T$, where I is impact, P is population, A is affluence (or activity), and T is technology. It was proposed in the 1970s as a way to conceptualize human impacts on the environment.

The last item, "Fermi Problem"/IPAT calculations may not sound familiar, but it is in fact a very common approach. In this approach, calculations take the form $I = \sum_i D_i \times R_i$, where I is an impact, the D_i's are a driver variable, and the R_i's are ratios (for example, resource use intensities or pollutant emission factors). The drivers and ratios may depend on other variables. The main reason for using such a simple framework is that it is more transparent than other options and easier to defend. A related benefit is that it is frequently easier to obtain values for drivers and ratios than directly for impacts, whether from published projections, expert estimates, or as as extrapolations of historical data. Finally, it can offer a useful starting place for more elaborate models, which often have such simple relationships embedded in them.

Assigning indicators to modules

It is helpful to assign each of the indicators produced by the model to a single module for planning purposes. This should be done whenever possible. However, it is not always possible to assign indicators unambiguously, since the calculations for a particular indicator may be spread across several modules.

Scheduling and Estimating

When developing a custom scenario model produced in the context of a particular scenario study, there are advantages to taking an *iterative* and *incremental* approach to development. In this case, versions of the model are released at intermediate stages – the iterative aspect – and improved and elaborated with each iteration – the incremental aspect. One benefit of this strategy is that it takes advantage of the availability of the experts who have been gathered for the scenario exercise, while another benefit is that the experts and stakeholders can get some inputs into their work from initial versions of the model. That is, an iterative and incremental strategy permits a dialog between the experts and stakeholders and the model as it is developed.

Time estimation

The first step in estimating the time required for model development is to decide on the number of iterations. Typically, two or three iterations is sufficient. The second step is to estimate the time for each module in each iteration. The estimate should take into account the necessity of research, data collection, and documentation, in addition to the core tasks of designing and implementing the module. This can be done by following these steps for each iteration:

1. Estimate how long it will take to decide on a module structure and implement it within the chosen modeling tool (without including data collection, research, or documentation), according to the best judgment of the people responsible for building the module.
2. Estimate the total time for module *i* using the following formula[5]

   $$T_i = d_i\left(3 + \frac{N-1}{6}\right)$$

   In this formula, T_i is the total time to implement the module, d_i is the time estimate for designing and implementing the model, and N is the total number of modules. The formula takes into account the time required for research, data collection and processing, documentation, and implementing interactions between modules.

5 The coefficients in the formula are based on data collected over the course of several projects at SEI.

The total time estimate for the project is then the sum for all the modules across all of the iterations. It can be converted to calendar time by dividing by the full-time equivalent effort available on the project. (For a small project this can be estimated by simply adding up the full-time equivalent of the people building the model. For example, 2 full-time people and one half-time person would amount to 2.5 full-time equivalents.) The time estimate can then be used to generate a budget.

For the first step, professional estimators have found it useful to use increasing increments of days for estimating time, rather than trying to be precise. Common choices include powers of 2 (1 day, 2 days, 4 days, 8 days, ...) and the Fibonacci sequence (1 day, 2 days, 3 days, 5 days, 8 days, ...). Another possibility used at SEI is to use 1 day, 2 days, 1 week, 2 weeks, or 1 month (that is, 1, 2, 5, 10, or 20 work days).

Dealing with "sticker shock"

A common reaction after developing a budget estimate is "sticker shock," meaning that the estimated time and budget is much higher than first anticipated. The term "sticker shock" comes from the experience of car buyers in the U.S., to whom car salesmen will suggest a number of amenities. Each amenity costs a small amount by itself, but when the customer sees the total on the "sticker" he or she is shocked.

A natural response to sticker shock is to want to change the estimate, but this should be resisted. An estimate should follow from a reasonable assessment of the amount of work to be done, backed up by data collected on past projects, so if the estimate exceeds the budget, it should be taken as an early indication that the project may be too ambitious, and that the scope should be reduced.

In some cases the scope of a project can only be reduced by removing indicators (and sometimes entire modules). The first to be cut should probably be those indicators labeled "could have" using the MoSCoW rules. If necessary, some of the "should have" indicators may also need to be cut. However, in many cases the scope can be reduced by using a simpler approach to calculating the indicators, or even by avoiding the calculations entirely by using outputs from previous studies.

One scope-reducing option that should *not* be taken is to omit documentation. Without documentation a model is almost useless. Only the people who will be interested enough to look at the actual model code will be able to assess it, so only they can potentially feel confident in applying the outputs.

Model Development

The details of each module and each scenario model will vary considerably, depending on the state of knowledge in the relevant fields, the time and resources available, and the expertise of the people involved. It is not feasible to cover all of the details of model building in a small handbook. Instead, this section will focus on the general tasks involved with model development.

Starting a module design

It is much more difficult to start developing a module than it is to continue developing it. The first step is challenging because few constraints have been placed on the work. To add constraints at the outset, it can be helpful to think about the goal of module design as shown in Illustration 2. As shown in the illustration, a module has inputs – perhaps some common drivers, or values produced by upstream modules. It also produces outputs – perhaps some indicators, or values needed by downstream modules. The goal when designing the module is to fill in the space in the middle, where the calculations happen, in a way that is both parsimonious and technically satisfactory.

One approach to starting the module design process is to write on a white board or flip-chart the inputs at the top of the white board or flip-chart page and the outputs at the bottom. Then the design session can focus on filling in the diagram for the intermediate module calculations.

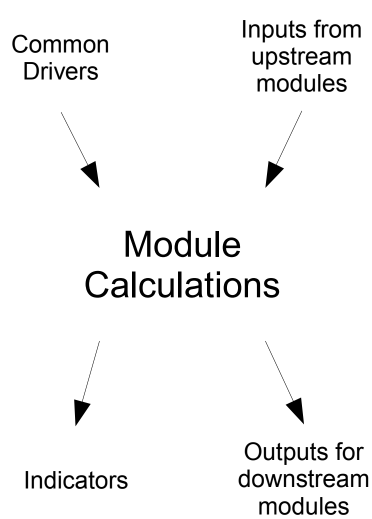

Illustration 2: The goal of module design

When designing a module it is important that the diagram be consistent with the modeling tools and model structure being used. For example, if the tools do not allow feedbacks to be incorporated in the model, then feedback loops should not be placed in the diagram.

Carrying out an iteration

In the iterative and incremental approach to model development advocated in this handbook, some activities tend to recur within each module and each iteration. These are:

- Design and research
- Data collection and processing
- Implementation
- Documentation

Additionally, either during module development or after, some time usually needs to be devoted to interactions between modules.

It is a good idea to think about, decide on, and prepare the supporting tools that will be used to carry out each of these tasks. For example, a white board may be used for design, a bibliographic database may be used for research, spreadsheets may be used for data processing, specific modeling tools may be used for implementation, and a particular word processor may be used for documentation. There may also be a need for tools to share information between several people working on the project.

Tracking progress

The goals for each iteration should be focused on some small and measurable unit, and only complete units should be counted when assessing progress. One possibility is to track progress by counting how many of the indicators are being calculated, although it is not the only possibility. If indicators are used as the metric, then they should be considered "complete" only when they are both implemented (up to the level expected for the current iteration) and documented.

It is highly unlikely that the work on an iteration will precisely meet the goals in precisely the estimated time. Instead, it will tend to be either over or under by some amount. When tracking progress, it is a good idea to either keep the goals for the iteration fixed, or the time for an iteration fixed. If the goals are fixed, then once the work is done, the actual time taken to complete the goals can be compared to the estimate. If the time is fixed, then once the iteration is done the amount of work actually accomplished can be compared to the work that was expected to be done under the iteration.

Estimated time or work products for each iteration can be compared with the actual time taken or work produced for each iteration and the overall progress tracked over time. If it is observed that goals are consistently unmet or exceeded, then the project schedule can be revisited and possibly revised.