

## Numbers in Scenario Studies

22 August 2004 (revised 2 October 2004)  
Eric Kemp-Benedict

### Introduction

While the heart of most scenario studies lies in the scenario narratives, formal analysis can add a lot to the process, by helping to clarify and sharpen the narratives and by providing illustrations of the scenario dynamics. Ideally, this should take place in an iterative process in which the narratives and formal models are developed together. This note will focus on formal *quantitative* analysis, in contrast to formal qualitative models (such those captured in causal loop diagrams). As narrative writers interact with a quantitative scenario model, there are essentially four different kinds of numbers the quantitative analyst has to keep track of - *data*, *parameters*, *inputs* and *indicators*. This note describes these different kinds of numbers and points out how IPAT-S and supporting software can help manage different numerical inputs and outputs.

### Information Flow

The basic flows of numbers and other information between the quantitative analyst (the “modeler”), the quantitative model and the model user are shown in Figure 1.

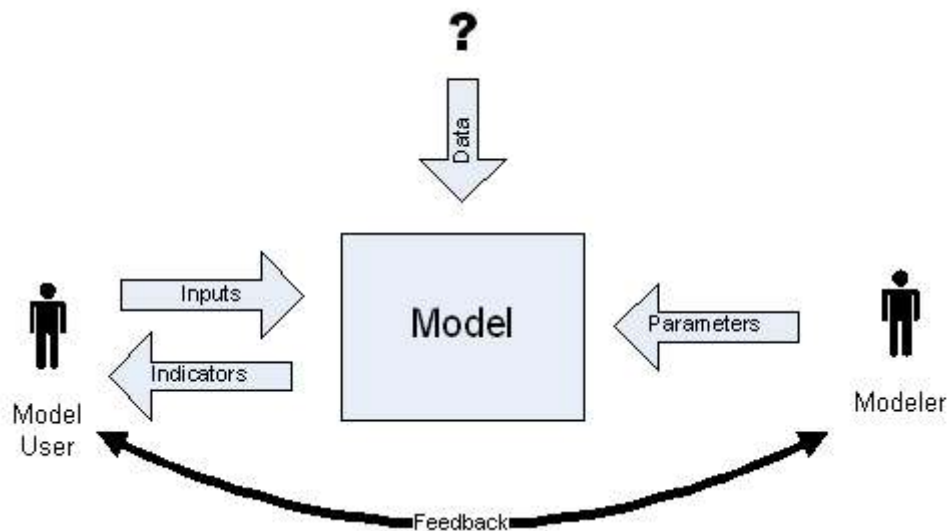


Figure 1. Information flow in a scenario exercise

As illustrated in Figure 1, the modeler and the model user communicate in two ways: directly in conversation, and indirectly via the model. In direct conversation, the modeler and the model user give each other mutual feedback on the plausibility of their aspects of the scenarios. The model user (ideally) is able to interact with the model by providing key inputs (for example, assumptions on economic

performance or technological change) and reviewing model output in the form of indicators. The modeler provides the model itself as well as the parameters of the model (that is, numbers that the model user is not expected to change) and the user interface. The data might come from the modeler, the model user or another source (e.g., support staff).

## Strategies for Managing Numbers

A serious scenario study generally involves considerable data gathering, data processing and model building (both qualitative and quantitative). The number of individual values for data, parameters, inputs and indicators can be extremely large, easily reaching into the thousands.<sup>1</sup> It is important to manage this carefully so as not to overwhelm the model user and the modeling team. This is particularly important for development and environment studies, because for such studies time and money budgets are usually very constrained. Some strategies for managing the large volume of numerical information are given below.

### *Inputs and Indicators*

The model user interacts with the model by providing key inputs and reviewing the indicators reported by the model. Both the inputs and the indicators should be kept to a minimum to reduce cognitive overhead for the model user. This is, in fact, one of the main purposes of using a formal model – ensuring that the model user does not have to keep everything in his or her head. For quantitative models specifically, selecting the appropriate inputs and indicators is a very tricky problem. In a “business-as-usual” scenario, it might be appropriate to build conventional economic assumptions into the model and not expose them to the user. However, for a scenario that is exploring radical alternative economic arrangements, such a model would be inappropriate. Also, if too much is hidden from the user, then the model user is apt to criticize the model as being a “black box,” even if the model is in principle easily accessible. For these reasons, the scenario modeler is often tempted to give the model user the greatest possible flexibility, but this temptation should be resisted. Whether the model user knows it or not, they would also not be happy to be given too many choices.

There is no simple solution to this basic problem. Indeed, it is almost guaranteed to arise. Aside from the central recommendation to keep the number of inputs and indicators small, it is also recommended that the scenario modeler adopt a philosophical attitude toward the demands of the model users. No matter what is presented to them, model users are apt to criticize. Moreover, their criticisms are almost always legitimate. If you ask potential users for assumptions “off the top of their head,” they will say (correctly) that they have too little information to go on. If you present them with a quantitative model instead, to help guide their input, they will say (correctly) that the model is too constraining. If you present them with many inputs and outputs, they will say (correctly) that the model is overwhelming. Finally, if you present them with a manageable number of inputs and outputs, they will say (again correctly) that the model is a black box and omits important possibilities. The fundamental problem is that humans do not have the innate cognitive ability to take in most environmental and social problems in their entirety at one time. In the process of generating narratives, building models and interacting with the models a sophisticated understanding can emerge. This is why the *process* of scenario development is so crucial, and often has more impact than the final product. (However, the final product can also have a strong impact if it is thoughtful and well written.)

---

<sup>1</sup>For example, if air emissions from fuel combustion are of interest, then within the industrial sector, in a typical study, perhaps 4 air pollutants and 5 fuels will be considered, for 10 industrial subsectors and 4 scenarios. In this case, there are  $4 \times 5 \times 10 \times 4 = 800$  figures for emissions and  $5 \times 10 \times 4 = 200$  figures for fuel consumption, giving 1,000 values for one theme for the industrial sector alone.

## Data

The availability of good-quality data often determines the limits of the quantitative analysis. It is important to get the best available data for the study, but it is easy for the search for data to get out of hand. Some general recommendations:

1. Only ask for the data that are actually required for the analysis (but do not limit the analysis unnecessarily by assuming that data are not available).
2. Whenever possible, ask for data in the form in which it appears in standard statistical reports, and automate any data conversions.
3. Be prepared to do a preliminary analysis using the most-available data, with the understanding that the entire data set might be replaced with locally-available data when the model is reviewed.
4. Collect data sets regularly.

Considering these points one by one, the first point is the key to limiting the potentially endless search for data. Make sure that whoever is gathering data knows what is required of them for the analysis at hand. However, be open to learning about unexpected sources of data. If the quantitative analysis is tailored to a generic data set, and the region under study has an extraordinary source of data that might be useful, then the analysis might need to be modified to take this new data source into account.

The second point, to use data in its original form, is important for at least two reasons. First, it reduces the burden on whoever is collecting the data. As this is often a client or a relatively inexperienced technical support person, this is an important consideration. You do not want to alienate your client, and you do not want to rely on inexperienced staff to do sophisticated data conversions. Moreover, as a programmer the model builder has a responsibility to automate the tasks that can be automated. Second, it makes it much easier to update the analysis when new data become available (and you should assume that you will *always* be asked to update the data). As an example, suppose that energy data from the International Energy Agency (IEA) of the OECD are being used to study industrial energy consumption. The IEA reports industrial consumption by highly aggregate industrial subsector groups. However, data on industrial value added (e.g., from UNIDO) are typically reported by International Standard Industrial Classification (ISIC) category. Different ISIC categories must be combined to get the IEA industrial groups. In your data request, ask for energy data by IEA grouping and value added data by ISIC category, and then have the model (or other) software do the job of combining the ISIC categories.

The third point is in part a suggestion to keep the analysis flexible. Again, you should assume that you will be asked to update the data set. This is very important to remember, because it is easy to cut corners with the data collection, for example by doing sums on a calculator and then entering the result directly into a model input file. When the (inevitable) request arrives to update the data set, it is difficult to trace the origin of the numbers and burdensome to go back and redo all of the sums. One particular reason that data are regularly updated, especially for a local study, is that local data are generally preferred to international data sets. Sometimes this is for purely political reasons, but mostly it is because of real problems with the international data sets (for example, because the international data sets use reporting categories that do not match those of the individual countries). Nevertheless, when initially developing a model it is often necessary to use international data sets. One reason for this is that it is difficult for those providing the data to really understand what is being asked of them without seeing a concrete example (although a well-designed data request can minimize this problem). Another reason is that waiting for initial data can seriously delay the start of model development. It is best to start with *something*, even if it is not the data that will be used in the final analysis. A final reason is that in many cases the international data are the only data available. In this case, it is good to be able to reassure the person responsible for data collection that some figures are available. Furthermore, they should *not* be asked to search for standard

international data sets. This should be part of the modeler's area of expertise. A modeler might provide data sets to someone else to do data entry, but that person should not have to find the data sets themselves.

This leads to the fourth point, which is that you should continually be looking for data sets. Collect data sets on your computer or a CD, and also collect web addresses for data sets. One source of online data sets for environment and development scenarios is the collection at <http://scenarios.kb-creative.net/>.

### **Parameters**

Parameters are the inputs in the model that you do not expose to users. For example, in a study of carbon emissions, the IPCC Level I emission coefficients might be put into the model. Taking a more controversial example, in a scenario that features conventional economic development, the model might include a neoclassical production function in which the user sets the rate of growth of total factor productivity, but the marginal contribution of capital to total output is fixed. In this case, growth of total factor productivity is an *input*, while the capital share in output is a *parameter*. The distinction between inputs and parameters is necessarily blurry, since the classification depends on the study. For example, in a scenario that departs significantly from conventional economic development, a neoclassical production function would most likely not be used at all, much less included with fixed parameters.

There are few general guidelines for managing parameter values, as they are tied closely to the model structure, and the question of how to develop appropriate models is very involved. However, two suggestions are: 1) to thoroughly document any parameters and the models in which they appear and, 2) wherever possible, to gather parameter values in one place. That is, if the modeling tool in use allows it, all parameter values should be placed in one part of the program, script or spreadsheet. This makes it easier to modify the parameters as the analysis evolves and easier for others to review the values.

## **Using IPAT-S and Supporting Software to Manage Numbers**

The IPAT-S scenario scripting language, IPAT Studio and IPAT DX are all designed to manage numbers following the framework in this note. For information about the IPAT-S language and the free (and open-source) IPAT-S interpreter, IPAT Studio graphical interface and the IPAT DX data management software, go to <http://ipat-s.kb-creative.net>.

### **Inputs and Parameters**

Key inputs and parameters can be managed in IPAT-S and IPAT Studio using IPAT-S syntax, in which key inputs are indicated by placing them in angle brackets, < >. The IPAT-S interpreter simply ignores the brackets, but they are used actively by the IPAT Studio interface. First, IPAT Studio highlights numbers in angle brackets in red, so they are easy to spot. Second, IPAT Studio offers keyboard shortcuts and toolbar buttons to jump from one input to the next. Third, and finally, IPAT Studio allows developers to create user interfaces that let users interactively change key inputs. This third possibility allows developers to distinguish the inputs that users should change (the ones exposed in the interface) from less important inputs and from model parameters. Less important inputs and parameters can be indicated using angle brackets to highlight them in the script, but need not be exposed through the user interface.

A good practice is to put all parameters into a single read file in the IPAT-S script. This keeps them isolated from the rest of the script while also gathering them in one place, allowing them to be easily reviewed and modified.

### **Indicators**

Indicators are of major importance in quantitative scenario analysis. A major goal of such an analysis is the production of a useful indicator set. Because scenario development is dynamic and interactive, the list of desired indicators can change rapidly. It is important to be able to quickly generate new indicators when a scenario team, stakeholder group or other model user needs them. For this reason, IPAT-S offers a quick

and flexible syntax creating indicators, the *report* statement. Once the indicator list becomes more stable, structured reports can be created using *print blocks*. For large studies it is recommended that a print block be used to generate a data set that can be read by the IISD Dashboard of Sustainability. The Dashboard can be invoked from within IPAT Studio, and is designed to navigate very large and complex data sets.

### **Data**

The IPAT DX program (DX stands for “Data eXchange”) is a tool for managing and exchanging structured data. It can also export data in a format that can be read into an IPAT-S script. It can greatly simplify the task of managing data for a scenario exercise. Mainly, it can be used to communicate a data request to someone collecting data for a scenario study. IPAT DX has a spreadsheet-like interface in which each variable appears in a different table. Variables are further grouped by category and, if needed, different IPAT DX data files can be used for a single study. For example, an energy data file, a water data file and a socioeconomic data file could each be sent to a different ministry, government department or expert.

Using a tool like IPAT DX, an IPAT-S analysis can be easily updated if the data set changes. The new data can be exported into IPAT-S form, read into the script and the script re-run. Also, it clearly separates data from the analysis itself. This makes it easier to review the analysis, since in a typical study, initializing variables with data takes up much more of the script than the analysis itself.

### **Summary**

There are essentially four different kinds of numbers a quantitative scenario analyst must manage – inputs, indicators, data and parameters. Each presents its own management problems, but there are techniques that can be used to make the job simpler. Some problems of interacting with model users are essentially intractable because of the limited cognitive capacity of humans and the complexity of policy-relevant issues. Nevertheless, understanding the source of these chronic difficulties can foster a philosophical attitude in modelers and help them to work more effectively with model users. The IPAT-S scenario scripting language and its supporting software is designed to help scenario modelers manage the different kinds of numbers they are required to deal with.