

Metamodeling for IPAT-S

by

Eric Kemp-Benedict

© 2005 Eric Kemp-Benedict

The layout for this document was designed by Mirto Silvio Busico
(m.busico@ieee.org) as an OpenOffice.org Writer template.

Index

1 IPAT-S and Metamodeling.....	2
2 A Metamodeling Language.....	2
2.1 Summable Variables.....	2
2.2 Ratio Variables.....	3
2.3 Share Variables.....	3
2.4 Reading the Diagram.....	4
2.5 Annotating and Evaluating.....	4
2.6 Turning a Diagram into a Script.....	5
3 Conclusion.....	8

1 IPAT-S and Metamodeling

IPAT-S is a scripting language (a special-purpose computer programming language) for quantitative scenario analysis. IPAT-S is in particular designed for sustainability scenarios.

As with most programming languages, it is best to plan the structure of a program before sitting down to write it. For most programming languages, this activity is referred to as “modeling.” However, for a modeling language like IPAT-S, this is a potentially confusing term, so in this How-To, the term *metamodeling* will be used instead. This How-To presents a diagrammatic approach to metamodeling that works well with IPAT-S. Although the approach presented here cannot be used for all the kinds of models that IPAT-S supports, it can be used for many studies and in nearly any study is a useful starting point.

All IPAT-S software and documentation is available for free from the IPAT-S web site, <http://ipat-s.kb-creative.net>.

2 A Metamodeling Language

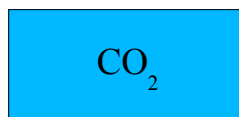
The key is to first identify the variables in the problem that correspond to *summable variables* in IPAT-S. Summable variables are variables that it makes sense to add up, such as total population, GDP, total emissions, etc. They contrast with *ratios*, such as GDP/capita, which are calculated by dividing one summable variable by another.

There are at least three reasons to focus on summable variables. First, with sustainability problems, the scale of an impact is usually expressed as a summable variable – total emissions, total resource use, etc. Second, expressing results in terms of summable variables makes aggregation much easier – if emissions have been calculated at a national level, then regional emissions can be calculated just by adding. Finally, most indicators can be calculated as a combination of summable variables, and again it is easier to construct aggregate indicators if they are expressed in terms of summable variables.

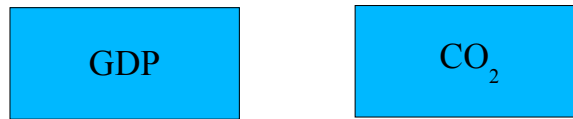
To make the approach concrete, a typical scenario calculation will be traced through. Suppose that the task is to create a scenario for national CO₂ emissions. There are several ways this can be done, so it is best to do a metamodeling exercise before committing to an approach.

2.1 Summable Variables

In the diagrammatic metamodeling language, summable variables are represented as square boxes. For the example of CO₂ emissions,



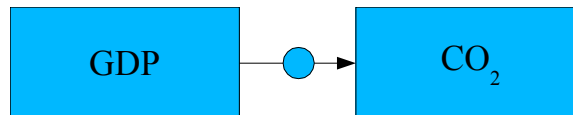
In a typical scenario calculation, carbon dioxide is thought of as being driven by something activity. At the simplest level, aggregate economic output (gross domestic product, or GDP) can be considered as the driver. So there is another summable variable, GDP.



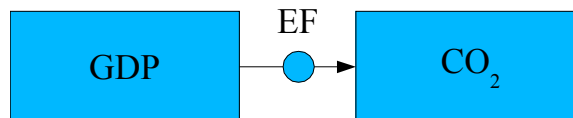
These variables must now be connected, to show that GDP is driving CO₂. This is the job of a *ratio* variable.

2.2 Ratio Variables

In the diagrammatic metamodeling language, ratio variables are indicated by circles. After putting down two summable variables, one of which drives the other, a ratio variable should always be inserted between them:



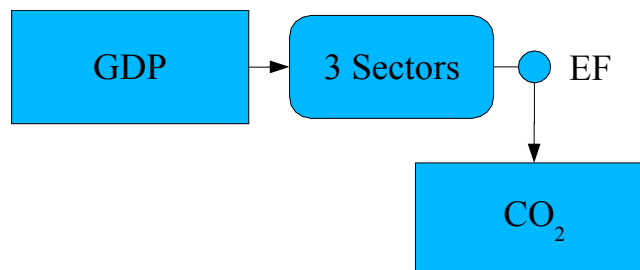
In this example, the ratio variable represents CO₂ emissions per GDP. This is an emissions factor, and the diagram can be annotated to show this.



2.3 Share Variables

A final variable type that falls into neither the “summable” nor the “ratio” category is a *share*. In IPAT-S, shares are represented as simple *variables*. A share is used to disaggregate a summable variable into components. For example, suppose that instead of looking at aggregate output it is decided to focus on sectoral output (primary, secondary, and tertiary). In this case, a separate emission factor could be introduced for each sector.

In the diagrammatic language, shares are represented by rounded rectangles.



This indicates that GDP is allocated into three sectors using shares, and that each sector has a separate emission factor and emissions level. The sectoral emissions can then be summed to give total CO₂.

2.4 Reading the Diagram

Once the diagram is constructed, it contains some valuable information that can be read directly from the structure. The rules are:

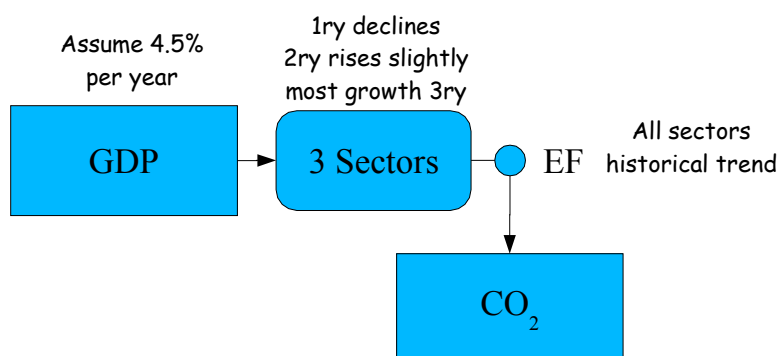
1. Every summable variable needs base-year data.
2. The first summable variable in a chain of calculations needs scenario values.
3. Every ratio variable must be given a scenario trend (e.g., growth rate or index).
4. Shares must be given in the scenarios, and they multiply how much data and how many assumptions must be made.

So, in the example diagram so far:

1. Both GDP and CO₂ (emissions by sector) require base-year data.
2. Scenario values are required for GDP.
3. Each of the three emission factors must be given a scenario trend.
4. Sectoral shares of GDP must be given for the scenario.

2.5 Annotating and Evaluating

As discussed in the previous section, the need for scenario assumptions can be read directly from the diagram. The assumptions themselves are added to the diagram as annotations. For example,



In this case, one annotation is specific, the other two somewhat vague but suggestive. To fulfill the “historical trend” requirement, historical data must be available. The sectoral shares must be made more specific, but since the requirements as they stand are not specific, this might be a possible scenario variable. That is, the scenario quantification team may offer several different values for shares to show the narrative writers the implications of different assumptions.

At this point the feasibility of the model can be evaluated. Data requirements and scenario inputs have been spelled out, either as an annotation or because they can be read off of the diagram. It

may also be decided at this point that the level of disaggregation is insufficient. (Although in general it is best to start with a simple approach first, and disaggregate at a later stage.)

A more thorough diagram might look something like the following (where for convenience different sectoral calculations have been grouped together in labeled boxes). With these two diagrams – and perhaps others – in hand, the scenario team can decide on a suitable strategy.



2.6 Turning a Diagram into a Script

It is relatively straightforward to go from a diagram to a script. This is one of the strengths of the diagramming approach. Using the simple example of CO₂ emissions driven by GDP that has been developed throughout this How-To, here is the procedure for creating an IPAT-S script.

First, specify the years (this is *not* in the diagram, but as scenarios are about changes over time, it is a necessary input). Suppose in this case that the base year is 2002 and that scenario years are 2012 and 2022. Then the first part of the script is

```
base year 2002
scenario years 2012 2022
```

Next, if there are any shares, specify the dimension (or dimensions) for which the shares are defined.

```
base year 2002
scenario years 2012 2022

dim sector "primary" "secondary" "tertiary"
```

Next, specify all variables. Summable variables in the diagram become *summable variables* in the script; ratios become *ratios*; and shares become *IPAT-S variables*.

```
base year 2002
scenario years 2012 2022

dim sector "primary" "secondary" "tertiary"

summable variable GDP CO2{sector}
ratio EF{sector}
variable GDPshare{sector}
```

Each of the summable variables must be specified in the base year, and scenario values must be assigned for several variables, as outlined above. For example,

```
base year 2002
scenario years 2012 2022

dim sector "primary" "secondary" "tertiary"

summable variable GDP CO2{sector}
ratio EF{sector}
variable GDPshare{sector}

GDP.0 = 1100 # billion $PPP
# CO2 emissions in million metric tonnes
CO2.0{sector = "primary"} = 40
CO2.0{sector = "secondary"} = 120
CO2.0{sector = "tertiary"} = 90

# Base year and scenario values
GDPshare{sector = "primary"} = 30%, <20%, 10%>
```

```

GDPshare{sector = "secondary"} = 30%, <32%, 35%>
GDPshare{sector = "tertiary"} = 40%, <48%, 55%>

# Trends based on historical performance
EF{sector = "primary"} = gr(<-0.5%>)
EF{sector = "secondary"} = gr(<-1.3%>)
EF{sector = "tertiary"} = gr(<-0.8%>)

# GDP grows at 4.5% per year
:: >>gr(<4.5%>)-> GDP

```

In this script, angle brackets have been used to indicate scenario assumptions.

As the (almost) final step, the key scenario calculation is entered. This is the part of the script that is most directly related to the diagram – an IPAT-S *chain expression* that reproduces the sequence of calculations in the diagram.

```

base year 2002
scenario years 2012 2022

dim sector "primary" "secondary" "tertiary"

summable variable GDP CO2{sector}
ratio EF{sector}
variable GDPshare{sector}

GDP.0 = 1100 # billion $PPP
# CO2 emissions in million metric tonnes
CO2.0{sector = "primary"} = 40
CO2.0{sector = "secondary"} = 120
CO2.0{sector = "tertiary"} = 90

# Base year and scenario values
GDPshare{sector = "primary"} = 30%, <20%, 10%>
GDPshare{sector = "secondary"} = 30%, <32%, 35%>
GDPshare{sector = "tertiary"} = 40%, <48%, 55%>

# Trends based on historical performance
EF{sector = "primary"} = gr(<-0.5%>)
EF{sector = "secondary"} = gr(<-1.3%>)
EF{sector = "tertiary"} = gr(<-0.8%>)

# GDP grows at 4.5% per year
:: >>gr(<4.5%>)-> GDP

# Main scenario calculation
:: GDP >> EF -> CO2

```

The final step is to report the results of the calculation using either a *print block* or a *report statement*. It is quicker to enter a report statement, so that is the approach that will be used in this example.

```

base year 2002
scenario years 2012 2022

dim sector "primary" "secondary" "tertiary"

summable variable GDP CO2{sector}
ratio EF{sector}
variable GDPshare{sector}

GDP.0 = 1100 # billion $PPP
# CO2 emissions in million metric tonnes
CO2.0{sector = "primary"} = 40
CO2.0{sector = "secondary"} = 120
CO2.0{sector = "tertiary"} = 90

# Base year and scenario values
GDPshare{sector = "primary"} = 30%, <20%, 10%>
GDPshare{sector = "secondary"} = 30%, <32%, 35%>
GDPshare{sector = "tertiary"} = 40%, <48%, 55%>

# Trends based on historical performance
EF{sector = "primary"} = gr(<-0.5%>)
EF{sector = "secondary"} = gr(<-1.3%>)
EF{sector = "tertiary"} = gr(<-0.8%>)

# GDP grows at 4.5% per year
:: >>gr(<4.5%>)-> GDP

# Main scenario calculation
:: GDP >> EF -> CO2

report CO2 as "CO2 (million tonnes)"
report 1000 * CO2{sector=sum}/GDP as \
    "CO2 per GDP (tonnes/million $PPP)"

```

As can be seen in this example, now that the CO2 and GDP summable variables are available, the results can be combined in different ways. In this example, the indicator *CO2* is reported by sector (since it has *sector* as a dimension), while *CO2 per GDP* is calculated as an aggregate intensity for the whole economy.

3 Conclusion

This How-To presented an approach to *metamodeling* – that is, to deciding on the structure of an IPAT-S model before writing an IPAT-S script. The benefits of metamodeling are the same as for

modeling in any programming language. It is easier to throw a metamodel away than a script. Meta-models are quick to develop and quick to replace, so many possibilities can be explored quickly.

The particular approach to metamodeling presented in this How-To has some additional advantages. First, each of the elements – summable variables, ratios, and shares – have direct analogs in the IPAT-S scripting language, so it is easy to take a diagram and turn it into a script. Also, data needs and requirements for scenario assumptions can be read directly off of the diagram. This makes it easy to assess the feasibility of a model.