Guide for Good Modelling Practice in policy support

Nikolic, Igor; Lukszo, Zofia; Chappin, Emile; Warnier, Martijn; Kwakkel, Jan; Bots, Pieter; Brazier, Frances

DOI
10.4233/uuid:cbe7a9cb-6585-4dd5-a34b-0d3507d4f188

Publication date
2019

Document Version
Final published version

Citation (APA)
Nikolic, I., Lukszo, Z., Chappin, E., Warnier, M., Kwakkel, J., Bots, P., & Brazier, F. (2019). *Guide for Good Modelling Practice in policy support*. Delft University of Technology, Faculteit Techniek, Bestuur en Management. https://doi.org/10.4233/uuid:cbe7a9cb-6585-4dd5-a34b-0d3507d4f188

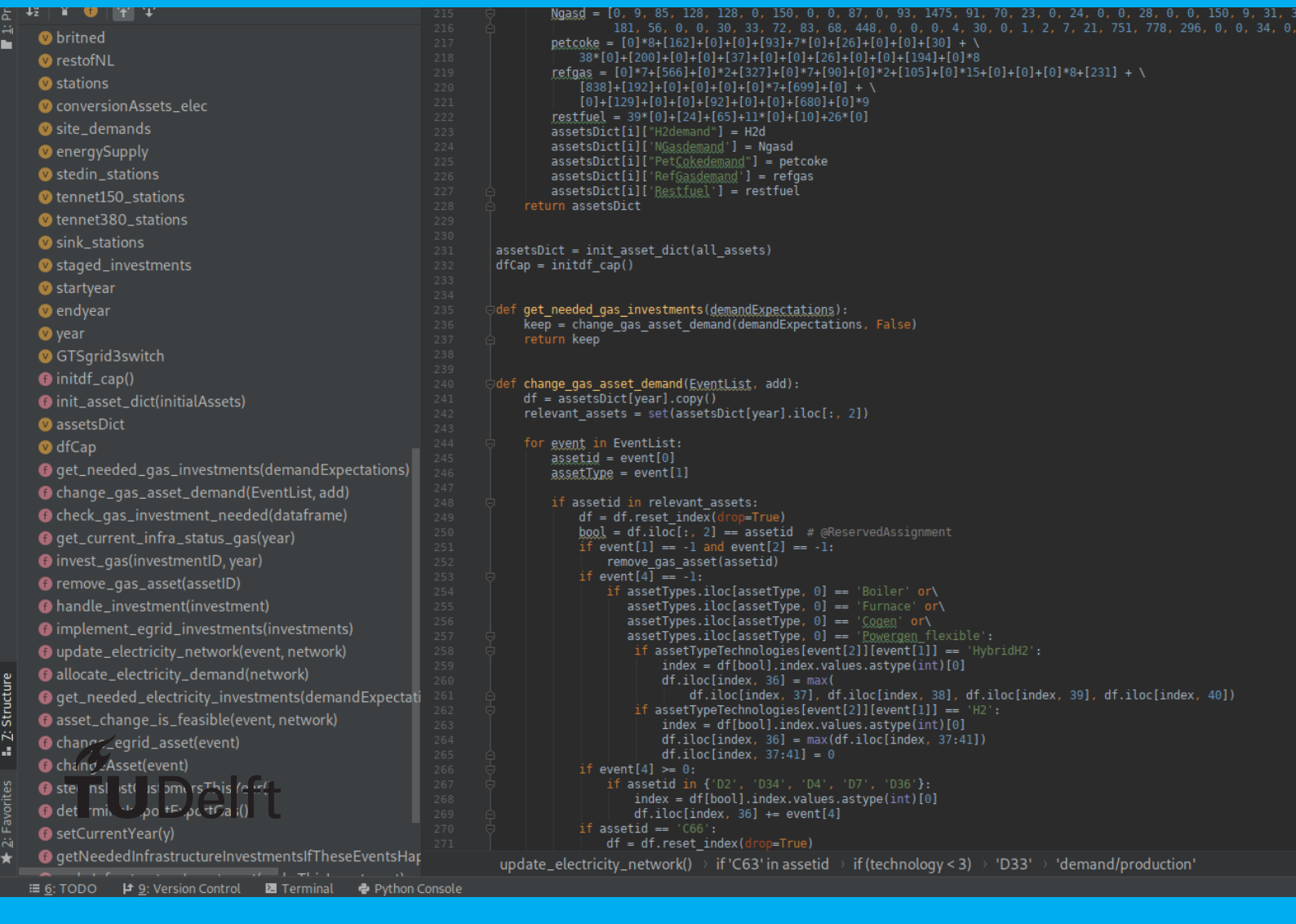
Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Guide for Good Modelling Practice in policy support

White-paper TSE7190007

17 July 2019

DOI:10.4233/uuid:cbe7a9cb-6585-4dd5-a34b-0d3507d4f188



Guide for Good Modelling Practice in policy support

Authors:
Dr.ir. Igor Nikolic, Prof.Dr. ir. Zofia Lukszo, Dr. ir. Emile Chappin
Dr. Martijn Warnier, Dr. ir. Jan Kwakkel, Dr. Pieter Bots, Prof. Dr. Frances Brazier

17 July 2019

White-paper TSE7190007 commissioned by

Rijksdienst voor Ondernemend Nederland
the Netherlands Enterprise Agency (RVO)

TOPSECTOR ENERGIE
Empowering the new economy
Topsector Energy

Number of pages: 26
Version 1.0
Electronic version available at
DOI:10.4233/uuid:cbe7a9cb-6585-4dd5-a34b-0d3507d4f188

Editorial responsibility and corresponding author
Dr. ir. Igor Nikolic
Systems Engineering and Simulation
Faculty of Technology, Policy and Management
Delft University of Technology, The Netherlands
Tel: +31152781135
Email: i.nikolic@tudelft.nl

TU Delft Delft University of Technology

Contents

1 Introduction **2**
- 1.1 How to use this document . . . 2
- 1.2 General principles . . . 3
  - 1.2.1 All models are wrong, but some are useful . . . 3
  - 1.2.2 Modelling is making choices and assumptions . . . 3
  - 1.2.3 The Garbage In = Garbage Out principle . . . 3
  - 1.2.4 R^5 Re-run, Repeat, Reproduce, Reuse, Replicate . . . 4
  - 1.2.5 Openness is essential . . . 4

2 Problem definition **5**
- 2.1 Identifying the policy question . . . 5
- 2.2 System boundary definition . . . 6
- 2.3 Modelling question . . . 6
- 2.4 Assumption tracking . . . 7
- 2.5 Technical requirements . . . 7

3 Model **9**
- 3.1 Choice for the modelling formalism . . . 9
- 3.2 Conceptual model description . . . 10
- 3.3 Model implementation details . . . 10
- 3.4 Model verification . . . 11
- 3.5 Model artefacts and limitations . . . 11

4 Input data **13**
- 4.1 Input Data selection . . . 13
- 4.2 Model parametrisation . . . 13
- 4.3 Input uncertainty handling . . . 14

5 Experimentation and results **15**
- 5.1 Validation . . . 15
- 5.2 Uncertainty handling . . . 16
- 5.3 Behaviour exploration . . . 16

6 Sense making and insight **18**
- 6.1 Output data handling . . . 18
- 6.2 Output analysis . . . 18
- 6.3 Output visualization . . . 19
- 6.4 Sensemaking . . . 20
- 6.5 Actionable insight . . . 20

7 Documentation **21**
- 7.1 User type relevant documentation . . . 21
- 7.2 Modelling project deliverables . . . 21

8 Way forward **23**

Bibliography **24**

1 Introduction

Models and simulations, particularly of complex situations such as the energy transition, water management, spatial planning, etc. are useful and powerful tools in supporting decision making. Creating them is hard and time consuming work. Making relevant and useful models is even harder and more time consuming. The rewards of such hard work can be great and real-world actions based on appropriate models can be vastly superior to acting on intuition or past experiences alone. However, if the model used or its outcomes are incorrect, we are worse off when we are not using a model at all. And since models are such complex and highly specialized tools, a non-expert in modelling might find it very hard, if not impossible, to identify a wrong, inferior or broken model, especially when it produces pretty pictures and desirable outcomes. Therefore, in order to provide model-based policy advice based on science, evidence, and data a practical set of guidelines for model definition, development and assessment is needed.

Modellers and decision makers often come from very different academic backgrounds and intellectual traditions, and might experience difficulties in communication. This document aims to help bridge this gap and help the decisions makers to commission and use models effectively and modellers to create correct, relevant, and useful models.

We further assume that the decision maker is interested in *what if, what is or how to?* questions such as ”What will be the uptake of electric vehicles if we implement tax cut?” or ”What is the best location of a wind park if we want to minimize noise exposure of our citizens and minimize the impact on the electrical grid stability?” or ”How to best support local clean energy initiatives?”. We assume the modeller is interested in creating a model or simulation to answer this type of questions in such a way that the answers are as useful as possible to the decision maker and that the model is an as accurate representation of reality as is necessary for the problem.

1.1. How to use this document

This document is not intended as an exhaustive and complete checklist of items, instead it summarizes the state of the art in Good Modelling Practice to the best of the knowledge of the authors. It translates these practices into specific questions the decision maker and modeller can ask each other in order to communicate needs and required information. These questions are meant to structure the conversation between the parties throughout the modelling project life cycle, align and clarify expectations and increase the overall quality and usefulness of the modelling effort. The policy maker can use this document to structure the content part of a tender, and the modeller to structure the model documentation and reporting.

This document has the following components:

Chapters Chapters describe the main steps in the modeling cycle that must be adressed for every model.

Sections Specific elements of the modelling process, and the activities that can be done to ensure maximum usefulness and transparency.

Oneliners Oneliners are catchy, tongue-in-cheek and easy to remember essences of a modelling step.

Guiding questions These are example questions that can be be posed during a conversation between the modeller and stakeholder to ensure that relevant aspects of that modelling steps are covered. If during a project all guiding questions are explicitly and satisfactorily addressed, one can be reasonably sure that the process is executed with all due diligence and has a good chance to provide a useful results.

Red flags Red flags are specific points that require extra attention. They describe classical symptoms of much deeper problems with the model and/or the modelling process, and if raised, are a cause for a closer investigation.

1.2. General principles

In this section the guiding principles are described that form the *spirit* of the Good Modelling Practice and may help in interpreting this document.

1.2.1. All models are wrong, but some are useful

The only fully correct model of reality is the reality itself. But even if we could build such a model, it would not be useful to us, as it is as complex as the real world [2]. Therefore we must *create and use* simplified representations of reality. ”All models are wrong, but some are useful” , a quote by George P.E. Box [3] perfectly captures the essence of modelling.

Therefore, every model is by definition wrong, since it is a simplification. But how to simplify? Simplifying too little means that we cannot understand the model, and therefore we cannot use it for answering our question. Simplifying too much means that the model is trivial and can not provide useful answers. Therefore, the goal of modelling is to construct the *least wrong, most useful model*, one that provides useful insights into the state of the world *and* can still be understood and interpreted.

1.2.2. Modelling is making choices and assumptions

As a consequence of the need to simplify reality, a model is really just a collection of assumptions with a ‘run button’. Which assumption and why is a choice made my the modeller and the decision maker. Often these decisions are made implicitly and are not explicitly articulated. Success of a modelling process and the usefulness of the resulting model are completely dependent on these choices and these must be made as explicit as possible.

Furthermore, the reader should be warned against a common misconception that that lots of details in a model means a good model. Overly detailed models may suggests exactitude, a closeness to reality that is not really there if the details are not relevant. A model that includes ”everything” to be ”as accurate as possible” is avoiding making choices and is thus not a good model.

1.2.3. The Garbage In = Garbage Out principle

Models are not magical future predicting oracles. They are machines constructed from systematic descriptions of how we understand the world around us, which convert facts and expectations into different facts and sometimes insights about the world. If we input wrong or irrelevant knowledge and/or wrong data, the data and insights that come out will also be wrong or irrelevant, see figure 1.1.

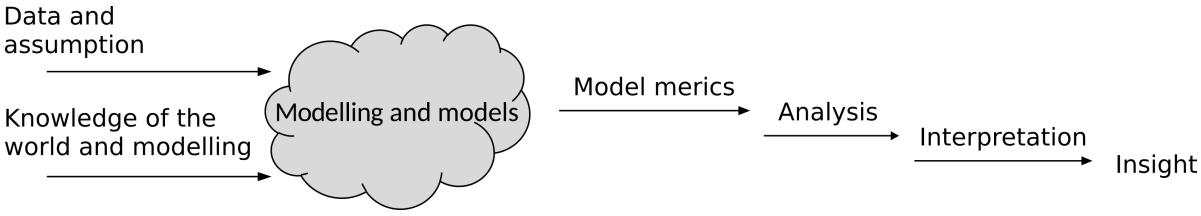


Figure 1.1: Inputs and outputs of a modelling process. The model is not a magical future predicting oracle but a chain of reasoning.

Every model therefore follows the *Garbage In is Garbage Out* principle

1.2.4. R^5 Re-run, Repeat, Reproduce, Reuse, Replicate

Given that models and simulations for decision support can have very large societal impacts, we believe it is essential to following the R^5 *approach: Re-runable, Repeatable, Reproducible, Reusable, Replicable* [1, 9]. In short this means that a model ought to be developed and used in a way that the results can be reproduced (re-runable), and that all the model elements are in place such that the analysis can be repeated, that the work can be (partly) reused in similar projects. These principles will not be discussed in detail here, but are fully integrated within the GMP guidelines.

1.2.5. Openness is essential

Models are conceptual and technically highly complex artefacts with potentially very large impacts. They are difficult to fully understand and check even for those that have the expertise and access to all relevant information and data. In situations where public resources are spent or public policy is made, it is of paramount importance to have the model and their data as transparent as possible. In view of this, the Dutch government, the EU and many other public organisations have adopted open source / open access guidelines [5] and created action plans [6]. The spirit of this document is that openness and transparency is essential in modelling.

2

Problem definition

The goal of modelling is to answer specific questions, not to provide general descriptions of systems. Asking ”Model the energy system transition and how our municipality can reach Paris climate goals” is not very useful, as a modeller has no way of determining what is relevant and what is not, and is therefore unable to draw meaningful boundaries and make relevant assumptions about how that system works. ”What is the impact of 10 wind turbines of 7MW each on the CO_2 emissions of our municipality, within the next 5 years, given a 10% rise of electric vehicle use per year?” is a question that a modeller can work with. It is a common situation that the question asked from the model is very different than the overarching concern of the decision maker. This means that it may take several iterations and significant time to identify it.

Therefore, when starting any modelling project, it is key to discuss and agree upon the following points:

- The overall policy question
- The modelling problem statement
- The system boundary definition
- Technical requirements
- Assumption tracking

If these points are not specified to satisfaction of the problem owner and the modeller, the project has very high probability of failure. It is possible that answering these questions is not immediately possible, as the process of modelling itself can help refine the stakeholders' and modellers' understanding of what is really the problem. This is fine, as long as it is clear that this is what we are doing.

Please note that this step is by far the most important one, and the one that is most often neglected. Sufficient time must be taken to develop the problem definition to the satisfaction of *both the decision maker and the modeller*.

2.1. Identifying the policy question

What is the reason for this model?

We start with exploring the general reason as to why we want to model something. What is the goal? What are we trying to understand. Answering the following questions may be helpful:

Policy goal? What is the policy goal that the problem owner is trying to achieve?

Problem to solve? What is the high-level problem that needs to be solved, for which this model should provide a part of the answer to?

Why model? Why do we believe a model is going to help us? Who is the intended model user?

Whose problem? Who is the direct problem owner? Who else has direct influence on the problem? Indirect?

Who is affected? Who is impacted by the decisions taken? Positively? Negatively?

Who models? Who is the modeller involved, and what is their relation to the other identified parties?

While these question can be very clarifying, they can also conflict with the politics of decision making, by removing useful ambiguity. By being very precise, we might find out that we actually do not agree.

2.2. System boundary definition

What is the system and its boundaries?

Given the overall problem statement and involved parties, we need to make decisions on the system and its boundaries. What is in, and what is not? Be aware that as the modelling process develops, we might have the desire to change system boundaries, to accommodate new or refined questions. Changing system boundaries while the model is constructed is incredibly disruptive and may lead to highly inaccurate and skewed models, and should be avoided as much as possible. We suggest that an agreement is made on *must have* and *nice to have* elements that are placed within the system boundaries in advance. Only once the model performs well within the must have parts, should the nice to have components be added. The following guiding questions identify system dimensions that may be relevant:

Spatial limits What is the smallest and larges geographic element in the model? What do we consider ”the rest of the world”? Is the model mm to m scale? Street level to municipal borders?

Temporal limits What is the shortest time period the model should be able to describe? What is the longest timescale we must consider? Milliseconds to seconds? Years to decades?

Energy limits What is the smallest amount of energy we are going to consider? A 4W LED lightbulb or a 7MW wind turbine?

Mass limits What are the smallest amounts of mass we will consider? A family garden waste container? A ship full of biomass?

Social limits Who will be described in the model in detail, who else is considered ”the rest of the world”?

Technological detail What is the level of relevant technological detail? What is considered black box, and what needs to be made explicit?

Organizational detail Which parts of organizations do we need to make explicit? Individuals in municipal departments? EU as a single entity?

Legal and institutional What the is the legal situation regarding the problem? Which habits, norms, shared strategies and formal arrangements are relevant?

A useful rule of thumb for setting boundaries is ”One level up, one level down” principle. The thing we want to study should be described from one level of detail more that then thing itself and that its effects should be studied at one level of aggregation higher. If we want to understand the behavior of a firm, we must model the people it consist of and the market it is a part of.

2.3. Modelling question

I want a model for this!

Once we have clear system boundaries, we can formulate the modelling question, as a very precise sentence using the XLRM framework, presented in figure 2.1.

Policy Levers These are the actions that we can control. The alternative decisions we can take the policy options available to us.

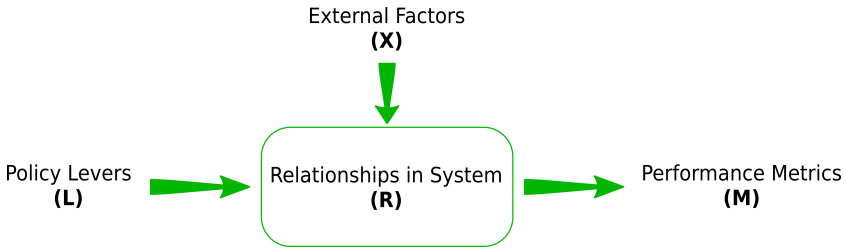


Figure 2.1: The XLRM framework [4]

Relations in System The system model that explicitly describes the relations between elements in the system, within the system boundary.

External factors All of the things that have influence on our system, that are not under our control. Things that are, can or will happen.

Performance Metrics The outcomes of interest. How are we going to express the impact of our actions? What are we going to measure? How are we going to measure it?

The modelling question can therefore be formulated as : **What is the impact/effect/consequence of *L* on *M* given *R* and *X***

Absence of a clear and mutually agreed modelling question is a key red flag!

2.4. Assumption tracking

Lets assume that....

As discussed in section 1.2, assumptions are the bread and butter of modelling. If one cannot know and understand the assumptions that went into the model, one cannot understand the model.

Every model development process should therefore be accompanied by a living documents containing list of all assumptions made. Each assumption should have a unique identifier, which is consistently used when presenting the model description, used in software source files to document specific implementation details, when analyzing and interpreting the outcomes and when specifying recommendations based on the model outcomes. Next to the description of *what the assumption is*, it is essential to document the reason why *this assumption is made*, even if, and especially when, it is ”we have no idea, so we just guessed”.

If this documentation process is started at the beginning of the modelling process, it requires minimal overhead and will pay back itself many times over in transparency and consistency. It will become invaluable at the end of the modelling process, when we have to explain the reasons and mechanisms of the model outcomes.

2.5. Technical requirements

Engine room calling the captain!

Next to the content and domain specific aspects of the model, there are always a number of technical requirements and limitations that the model has to meet, in order to be useful to the stakeholder. Answering the following questions may help identify these requirements:

Does it already exist? Organizations often suffer from *Not invented here* syndrome, where perfectly suitable models developed elsewhere are not used. Is there a suitable model already out there that can answer out modelling question? If yes, why are we not using it?

What is the desired performance? Does the model need to provide an answer directly? Can it take hours or days to provide an answer?

What are the hardware requirements? Must the model run on in-house hardware where a High-Performance computing facility may not be available? May it be operated ”in the cloud”, on hardware owned by commercial companies in a different country? Must it run on a laptop without internet connection?

What are the software requirements? Does the model have to run on a specific tool/platform/programming language that is already used? Is non-open source software acceptable? Must the model read / produce specific data formats? Should the model expose an API, allowing other software to use it? Must is interact with other software? Which?

Who is going maintain the model and how? Does the model need to operated for a long period of time? Will software remain accessible? How are we going to deal with updates?

3

Model

In this chapter we identify five specific elements of this step in the modelling process that every modeller should explicitly execute and communicate about. They are:

- Choice for the modelling formalism
- Conceptual model description
- Model implementation details
- Model verification
- Model limitations and artefacts

Elements discussed here are key parts of the models documentation, discussed in chapter 7. They are however so important, that they are discussed separately here.

3.1. Choice for the modelling formalism

Why are we modelling this way?

Every model is a simplification of reality. The first part of that simplification comes from system boundary selection and assumptions, discussed in section 2.2. The second part is the choice of the mathematical ”language”, or modelling paradigm, that is used to describe the relationships in the system. There is a large number of different modelling paradigms and systematically discussing all of them is outside the scope of this document. Here we present a incomplete list of some of the more common formalisms as an illustration for the non-modellers:

Statistical model Based on a (many) observations, a mathematical function is ”fitted” onto the data. Accurate description of ”what is” but impossible to use outside the data used. E.g. Econometric models.

Linear model Based on (sets of) linear equations. They are easy and fast to make, can not describe dynamic change. E.g. LCA models of environmental impacts, input-output models in economics.

Dynamic simulation Various ways to describe how a system, or its components behave over time, describing paths of how a system gets to its end state. Very sensitive to initial conditions and described system structure. E.g. System Dynamics, Agent Based Modelling

Technical design and performance These are models based on laws of nature and are used as predictions of specific technical or natural systems. They can be highly accurate, but are very strictly limited in scope of what they can model. Examples include electricity grid flow models, groundwater flows models, flooding models, etc.

Data driven model Models fitted to rich data sets. These models can make very accurate predictions, but are almost impossible to understand why and how. They are completely and critically dependent on the data set that they are trained on. Machine learning algorithms and AI models are prime examples.

Optimization Optimization is not a formalism itself, bur rather an approach to identify the ”best” solution with respect to a single or multiple criteria. It can be applied to models expressed in any formalism. What best means is completely dependent on the defined criteria. E.g. identifying best cost-performance options.

None of these approaches are good or bad, correct or incorrect by themselves. They are tools that can be used in a correct or incorrect way, can be suitable or unsuitable for the problem at hand. The modeller must be able to clearly explain *why* a particular type of modelling paradigm is chosen, and *how* it is suitable for the modelling problem. This prevents the ”When you have a hammer, everything looks like a nail” mental lock in.

Inability or unwillingness to explain the choice for a modelling formalism is a important red flag.

3.2. Conceptual model description

Once upon a time, in a land far, far away...

Once we are clear on which formal approach will be used to describe the system, the modeller should describe the actual model, i.e., the *R*elationships described in 2.3.

The goal is to ensure that all stakeholders are able to understand the model. Since a model is a highly specific simplification of reality and is not present in the real world, the stakeholders natural intuition of how the world works is not applicable for the model. Therefore, the modeller should enable the stakeholder to develop a new intuition abut how the synthetic, simplified world in the model works. A narrative form, the story of the model, is an excellent way to do so. it The following questions can be used as a guideline:

What does it do? What are the *R*elationships in the model. What is going on? What does it do, how does it do it?

How does it do it? How are e*X*ternal factors described and how do they effect the *R*elationships?

Which buttons can I push? How are policy *L*evers described and how to they influence the *R*elationships?

What do we measure? How are *M*etrics determined?

The description at this stage must be human readable and accessible to the stakeholder, preferably in a narrative, qualitative form. Flow charts can be used to precisely describe the models logic and flow. Where necessary, detailed quantitative mathematical representations can be used. In cases where the stakeholder is not comfortable with a mathematical notation, these details should be presented in the implementation details step (described below) instead.

3.3. Model implementation details

Details matter. A lot.

In this step the modeller presents the detailed description of the model. The goal is to provide an expert understandable description, down to mathematical/software primitives. The level of detail must be sufficient that another modelling expert can *Reproduce* the model independently. Even if the stakeholder may not have the modelling expertise or domain knowledge to fully understand the model details, they must have the means for independent review. In practice this can mean providing the full source code of the model.

As a general principle, a publicly funded modelling project should use open-source, publicly available tools and engines where possible, rather than closed source and proprietary options. Open source allows review and changes to those black boxes, if the need arises.

3.4. Model verification

Did we build the thing right?

Even the simplest models and simulations consist of many equations and algorithms as well as their implementation in many lines of computer code. As these equations and algorithms are created by humans, *they will contain errors and bugs*. These range from obvious ones, which makes the model not work at all, to very subtle logical mistakes, that none the less may cause the model to produce erroneous outcomes.

Is is therefore of paramount importance to verify the implemented model. Verification answers ”Did we build the thing right?” and is different than validation which answers the ”Did we build the right thing?”, discussed in section 5.1.

The modeller should provide evidence that the conceptual model presented in section 3.2 corresponds with the model implementation which produces the outcomes. There are several activities that can be used as part of the verification process:

Debugging What was the debugging process during model development? Is a software debugger used? Is there a issue tracking system linked to the version tracking system?

Unit testing Has the modeller used ”Unit testing” to demonstrate the correctness of the smallest logical elements in section 3.3? How complete is the coverage of the unit tests?

Continuous integration and testing Is every change to the model code tested against software regression errors? Are model breaking changes identified as soon as they are made?

Standard dataset results If standardized datasets are available, what are the model outcomes? How different are they, and can this be explained?

Extreme value testing How does the model perform when extreme parameter values are used? What happens when variable have positive or negative infinity values? What happens when values are 0? What are the parameter values where the model stops producing meaningful outcomes?

”Smell test” When presenting a base case / standardized dataset, does a model produce outputs that a domain expert finds consistent and explainable. If not, why not?

Pedigree Does the model (component) have a history of use in other cases, which vouch for its correctness?

While it is practically impossible to prove the complete correctness of a computer program, systematic verification can greatly reduce the number of errors, especially when its limitations and artefacts are systematically described.

3.5. Model artefacts and limitations

Sometimes it acts a bit strange...

A model artefact is model behavior that is not caused by the model *R*elations but is a consequence of mathematical or software peculiarities of the specific tool and / or implementation. For example, older versions of Microsoft Excel could not have more than 256 columns and 65536 rows, so any model that crosses that limit will have artefacts.

Modeller should, for each known artefact, at least report the following:

- A human interpretable name
- Description of the artefact
- Impact of the artefact on the model results
- Model parameter values where it occurs
- Mechanisms causing this artefact

- How to remove / reduce it, if possible

A model limitation occurs when specific parameters have values where model relations and assumptions are not valid anymore, or create unacceptable errors. For example, efficiency of a energy converting unit is higher than 1 or a price is negative. The goal of reporting limitations is to enable stakeholder to correctly interpret the outcomes and develop an intuition when the model can, and more importantly, cannot be used.

4

Input data

Knowledge and data are two key inputs into models, as discussed in section 1.2.3. So far we have dealt with the knowledge inputs. In this section we will discuss the various aspects that are involved in using data in models. A key starting point is the notion of FAIR (Findable, Accessible, Interoperable, Reusable) data principles [8], which the Dutch government has embraced [5].

4.1. Input Data selection

Which data?

Given the choices for the modelling problem statement, the choice for modelling formalism and the conceptual model, which data are needed for the model may drastically differ between models. The following questions may be used to identify them:

Which data is needed? Which specific data points does the model need? At which aggregation level? In which format?

Which data is available? What data is available that comes closest to what is needed? What needs to be done to get the correct aggregation level? What are the formats available, and how can they be converted to the needed format?

What are the restrictions on the data? Is this publicly available data that is free to use? Is it commercially available? Are there privacy issues? If so, how will they be dealt with?

What can be done to get missing data? Can we collect, generate or synthesize the missing data ourselves? How? When? How will it be made available?

Which part of the data? In some cases there might be too much data to use in the model. How are we going to select which subset if going to be used? Why this particular subset?

4.2. Model parametrisation

Where in the data?

Once we have identified and acquired the data set needed, we need to decide how to use it to parametrise the model, i.e. set the models variables to values that represent something useful. While this may seem trivial, if these points are not addressed explicitly, large number of implicit assumption can creep in the model and an greatly influence the model outcomes and their interpretation. The following questions can guide the parametrisation process:

Which state of the world? Are we describing the current situation? A historical one? A desirable future one? Or a undesirable one?

Why this particular data point? There usually are many alternative and similar ways to represent the desired state of the world in a model. Why this particular one?

What will be variable, what is set constant? Which of these model variables, parametrised this way, will be kept constant? What will be varied? Why?

4.3. Input uncertainty handling

There are known knowns, known unknowns and unknown unknowns. D. Rumsfeld

Data is per definition uncertain, due to measurement error, choice of measurement tool, the choice of what to measure, to name just a few. It is of paramount importance that every model explicitly handles uncertainty, both on the data and model structure itself. There are a number of questions that can be asked to ensure correct handling of input data uncertainty:

Margin of error Per variable, describe the estimated margin of error. Note whether the margin is estimated or measured.

Symmetry in uncertainty Per variable, describe whether the uncertainty is (a)symmetric, whether an optimistic or pessimistic estimate is used.

Expected developments Per variable, describe whether the data is inherently uncertain, whether the uncertainty may be reduced over time because more/new data becomes available, and what kind of developments may affect the margin of error.

Not explicitly addressing uncertainty in the input data is an important red flag!

5

Experimentation and results

With an experiment, we mean a run of a model with a specific configuration of the XLRM elements, levers, parameters, model relations and external factors. Result are all the model outputs, be it graphs, numbers, text etc. In this section we will discuss how to design the most useful computational experiment and how to systematically use these experiments to explore the model behavior and get rigorous results that can be used to answer the modelling question, and subsequently the overall policy question.

5.1. Validation

Did we build the right thing?

Validation concerns the fitness for purpose of the model we created. The criteria for validation should be identified and agreed upon before model experiments are performed. Validation is a surprisingly deep and nontrivial aspect of modelling, with a large body f scientific literature available. We will however reduce model validation it to four main ways relevant for us:

Classic validation How well does the model represent reality? In natural sciences one makes a model, which produces a prediction about how the world functions. Then, an experiment is performed and the model predictions are compared to them. The degree of mismatch is a measure of validity. Some policy related questions are of this type, e.g. questions about the performance of a technology. However, most of the policy relevant questions cannot be validated this way. Since such models often describe social systems, they developed from a specific subjective and normative perspective. Also, often they describe systems that do not exist yet. E.g. a model that explores the impacts of increase of subsidies for individual solar panels will be deemed invalid by a person whose political views reject distortions to the pure free market. regardless of the outcomes. A model exploring the effectiveness of a new marked design for local distributed energy generation cannot be validated in a classical sense, as such a system does not exist yet. In these cases we have the option of expert, or face validation, or historical replication.

Expert or face validation means that a group of experts examines the model and its outcomes, and forms an opinion on how good the model is. In answers the question ”Does the model look right” This form of validation is easy and fast to perform, but is extremely sensitive to the so called *selection bias*. Our choice of experts can be strongly influenced by whether they agree with us or not.

Historical replication means that the model is set back in time, and fed historical data. If the results match the historical record, the model is assumed to be valid. The main problem with this type of validation is that the data may not be available, or that the current reality has so much changed from the historical situation, that the model cannot (and should not) replicate it.

Validity through use Finally, the most pragmatic way to validate the types models most commonly build for policy support is to answer the question ”Is the model suitable for answering the question is has been built for?”

There are several questions that can be asked to explore models validity:

What is the validation strategy? How has the modeller validated the model? Which approach has been used? Which parametrisation? Which experts?

What is the deviation? How much do the model outcomes differ from reality / history / expert judgment? Why do they differ? Does it matter?

Is the model suitable for its intended use? Do we believe the model is fit for purpose? Why? Why not?

5.2. Uncertainty handling

Where are the error bars?

Models containing stochastic elements produce variable outcomes, in the sense that the outcome of different model runs are different and all those different outcomes are reasonable. The main causes are uncertainties in the input data, uncertainties in the model structure / mechanisms and artefacts due to internal model randomness. There is no way to completely eliminate this uncertainty, and it is therefore essential that this is dealt with explicitly. The modeller has to clearly communicate how uncertain the outcomes are, with respect to what they are uncertain, and what mechanisms, variables and parameter values are causing the most uncertainty. If the result uncertainty is as big or larger than the range of outcomes of interest, it means that the model is not suitable for the task it was built for. For example, if our model levers lead to a improvement of 20%, and the confidence band of the outcomes is 60%, our results are meaningless.

The modeller should describe an uncertainty analysis plan, identifying the steps taken to explore the model uncertainty and explicitly discuss their impact on the model outcomes. PBL provides an excellent”Guide for Uncertainty Communication” [7] which we suggest is followed. The following types of experiments can be performed to explore the uncertainty:

Model variability If an experiment is executed multiple times with identical parameter setting, does it produce the same outcomes? Should it produce identical results? If not, how different are they? What is the cause for this difference? Can the differences be explained on the basis of the stochastic elements in the model?

Input sensitivity testing Perform a sufficiently large exploration of inputs (X and L) around values of the base case (”reality” parametrisation). This leads to an understanding of the consequences of individual uncertainties. Do the outcomes, measured in terms of performance metrics (M) change less or more as the deviations in the input? Can this be explained?

Multi-variable sensitivity testing Systematically explore the multidimensional, multi-variant sensitivity analysis of the model, for example using Patient Rule Induction Method (PRIM). This gives insight in which combinations of inputs lead to particular outcomes (e.g. which combinations of inputs lead to high values for a particular performance metric). This provides an understanding of what combinations of parameters lead to either realistic or unrealistic results, or desired or undesired system behavior.

Not explicitly discussing the parameter (X and L) sensitivity of the performance metric (M) is a red flag!

5.3. Behaviour exploration

What does it do and why?

From this point on a ”cascade of sense-making” starts. This means that now the model development is finished, we will start exploring and understanding *what it does* and *why does it do it* in a series of steps. If we cannot make sense of model outputs in any of the steps, the entire cascade collapses and the models usefulness is greatly diminished.

In this step the computation experiments are described that will produce outputs relevant for answering the modelling question. Next to answering the specific question, the experiments are meant to explore the behavior of the model across a wide parameter space. This helps understand the models behavior and helps develop an intuition about how the model outcomes should be interpreted. The following questions can be used to set up model experiments:

Model levers What are the experiments that describe the specific model levers (L) we wish to explore?

Behavior exploration What is the full range of relevant input parameters (X), and how are the experiments going to explore the parameter space in relation to the levers that are of interest? What is a manageable number of computational experiments?

Statistical significance How many repetitions / experiments need to be run, given the known uncertainties, in order to get a statistically significant answer to the the modelling problem?

Not explicitly varying parameters (X) within reasonable ranges in experiments when testing the effect of a lever (L) is a red flag!

6

Sense making and insight

"Forecasting is very difficult, especially about the future." Niels Bohr

Now that we have identified that the model is valid, have mapped the uncertainties and performed relevant experiments, we are faced with (overly large amounts of) data. How can we convert long series of numbers into something we can act upon?

6.1. Output data handling

What comes out?

First order of business is to make sure that the data produced by the model are handled in a way that will allow us to trace from which experiment the data comes from and how the data was generated. We also need to make sure that we do not lose the result data, as in some cases rerunning large computational experiments can be very costly in terms of time and money. If not done properly, it may prove very hard to reproduce the model conditions. The questions to ask in this step are:

What is the data? Is there a systematic way to keep track of meta-data, the data describing the data? Which specific model run did produce this data set? When? By whom? Using which supporting software and hardware? Which model version, parametrisation and other input data were used?

Where is the data? What is the format of the output data? Can it still be accessed in the future? Where is it saved and backed up?

6.2. Output analysis

What do I see?

We have run the model many times, it has produced the data, it is clear what it is and is safely stored. Now what? Data analysis is the process of figuring out what the data can tell us. It is important to emphasize that during this step a important cognitive bias is at work, namely confirmation bias. Humans tend to believe facts that fit their preexisting ideas about how the world works or are positive for them more than negative ones. The reverse is also true, if the model outcomes are undesirable for the decision maker, one is far more likely to criticize the model, search for flaws and identify reasons to reject the output. In order to prevent cherry picking and blinding ourselves, the following questions can be asked during the analyses phase:

What are we looking at? Which output metric are we examining? Is it aggregated? How? Why?

What are we looking for? Are we identifying end states or specific solutions? Are we looking for changes over time? Some specific type of response? The lack of response?

What do we want to see? We identifying the effect of the levers on the model outcomes. How are we going to see them? Changes in what? How? When? Compared to what?

Where are we looking? What are the parts of the input parameter and results space that we are exploring? All? Some specific parts? Why?

How are we looking at it? What is the way we are transforming, aggregating our data? How are we defining and implementing differences? Is the processing repeatable?

How does it change? When / where do the outcomes change in a relevant manner?

What is unexpected? If something was unexpected, can we identify possible reasons why they are the consequence of an error in the model or is a counter-intuitive result that was not anticipated? Can we search for arguments to decide which is true?

If some of the outcomes can not be adequately and convincingly explained in terms of 1) the assumptions made, 2) the properties of the model and model type, and 3) the experiment, this is a **key** red flag!

6.3. Output visualization

How does it look like?

Once the analysis has been performed and the output data have been processed, the next step in model analysis is to visualize the data. It is essential to realize that there are three key cognitive biases in data visualization. They are:

Pretty = True The first bias occurs when very visually appealing graphs, chars and plots are presented, especially if they are animated. Users are far more likely to accept model outcomes from a pretty graph. This is not to say that visualizations should be unappealing, but one has to be aware of this bias, especially as it is very easy to abuse.

Too much = False The second bias is that when we are confronted with too much data, we tend to reject the evidence in front of us. Especially when many experiments are performed over many parameter values, visualizations can become overloaded with data, creating a visual mess and confusing the user, or lead to a ”graph barf” where hundreds of graphs are automatically produced, overloading the user. In these cases the stakeholder is much more likely to reject the outcomes, as they confuse rather than clarify.

Rare and special = True Finally, every model will contain outliers, artefacts and strange behavior on the edges of its validity scope, as discussed in section 3.5. Human minds have a very strong bias towards the out of the ordinary and unexpected. Therefore, when data is visualized, the user will tend to overly focus on ”that one weird run” and not so much on the bulk of the data that is ”just normal”. Especially if this outlier is very positive or very negative for them, this can lead to doubt about the model validity and usefulness.

The following questions can be used to identify a data visualisation strategy:

Message What is the key message that should be conveyed from the data? Is it consistent across all visualizations and metrics? If not, why not? If yes, how come?

Form Are we using the right metrics? Why has this specific visual format been chosen? Is it suitable for the audience and message? Which other formats are available? E.g. why a pie chart and not a bar chart?

Method How are the graphs produced? Is the method well described? Can they be replicated? Are all graphs generated ont he basis of results from the same model version? Are the analysis scripts available?

6.4. Sensemaking

What does it mean?

Now we have many graphs, trying to tell us a story of how our *L*evers effect the *M*etrics, though the *R*elationships under some e*X*ternal effects. Can we make sense of it all? Can we develop an intuition about how the modelled, simplified reality behaves?

The following questions can be used to make sense of the results, analyzed and visualized in a specific way by the modeller:

What is going on in this graph? Do we understand the message that this specific visalisation is trying to convey? Is it what the modeller wanted to convey?

Where are the artefacts and limitations? Do we see limitations and artefacts defined in section 3.5? How do we interpret the output given this?

What if I...? What are the impacts of the policy levers that the stakeholder controls? How to their impact look like?

Why is this going on? What are the mechanisms in the model that are responsible for this?

Through these questions we should be able to tell a story of the simplified, modeled, representation of the world, and have an reasoned explanation of every bump, point and line in the graphs.

Inability to answer the ”Why is this happening?” question is an important red flag.

6.5. Actionable insight

So, what should I do?

Finally, once we have identified the problem, described and created the model, gathered the data, run the model many times, analyzed, visualized and interpreted the data, the final question remains: What do we do now?

The model sense-making should have provided the stakeholder with a consistent intuition about how a simplified world in the model works, with a solid understanding of its limitations. We can now start reasoning about the real world, and argue for or against a specific action. The modeller is not responsible for the decision taken by the decision maker, but is responsible to warn the stakeholder if they are about to take it based on flawed reasoning about the model, its mechanisms, data or results.

7

Documentation

If the modeller and the stakeholder have systematically answered the questions posed in this document, there should already be a very complete documentation of the what, how and why of the model. This section will explicitly elaborate on several additional aspects of model documentation.

7.1. User type relevant documentation

Especially in cases where the stakeholders are going to use the model themselves, rather than just receiving the model results, relevant documentation for user training and education is key.

We can identify at least 4 different types of users, for which different types of documentation are needed:

Decision maker This user requires understanding of the high level concepts, intuition on why the recommended actions are formulated, and under which conditions model outputs are valid. One can think of the user manual for a machine.

Domain expert Specific model choices regarding the content, structure and assumptions of the model should be made accessible to this type of user, next to the insights provided to the decision maker. Here, the schematic and the list of parts of that machine is provided.

Software developer and maintainer For the users who are extending or changing the model, Aplication Programming Interface (API) and code level documentation is required. What is the conceptual model representation, what is the the model code architecture, what are specific implementation details and explicit links to domain assumptions. This is akin to providing the building instructions for the machine, next to the above descriptions.

System administrator The person providing the computing environment in which the model functions needs to understand the overall software and hardware environment the model operates in, its operating system and library dependencies, computational requirements, location and formats of code, data and results files. Here the instructions for building the factory that makes that machine are provided.

7.2. Modelling project deliverables

A modelling project meant to provide actionable insight to a policy maker in a complex situation, like the energy transition, is a non-trivial thing. In order to be of maximum usefulness, the following deliverables can be considered. They should be provided in open standard computer file formats next to physical reports. In case where intellectual property, data protection issues and software licenses are involved, suitable arrangements should be made, ensuring that the stakeholder can use and keep on using the model and its outcomes.

High level summary A high level advice, addressing the overall policy problem, based on the model and its outcomes.

Detailed report Extensive report detailing the model, data, results, and analysis.

Executable computer program Executable file that produces the results presented in the detailed report.

Input data All data needed to produce the result.

Output data All of the output data produced by the model run based on the input data.

Analysis scripts All code needed to produce the analysis and visualizations in the report.

Source code The entire source code of the model, preferably with its revision history

Executable full software stack A virtual machine image which can generate all of the results.

8

Way forward

In this whitepaper we presented a coherent set of best modelling practices for policy support. We have attempted to make the guidelines and the red flags as specific and concrete as possible. However, given the enormous diversity of questions, approaches, domains etc, they are necessarily fairly generic, and have to be adapted to the practical situation at hand.

They are not, nor can they be, a definitive and final set of rules that one must follow, and following them does not *guarantee* that the produced models are perfect, or even useful. However, when consistently applied throughout the model development cycle we will produce a model whose results will be of much higher quality and usefulness than without them.

Looking forward, we feel that if these guidelines are to be widely adopted in modelling for policy support practice, further work is needed along three themes.

First, the guidelines should be made even more practical and easier to communicate. A busy policy maker might not have the time to digest 20-odd pages of theoretical material. Brochures, infographics, ”10 steps to..” type documents can help communicate and use the guidelines.

Second, training critical uses is essential. Developing and using models is a non-trivial skill that takes a long time to master. We believe that developing training materials, such as online courses or training sessions on how to critically evaluate the problem definition, modelling process, use of model and how to prevent common pitfalls will be useful for both model users and model developers.

Third, there is a significant but unnecessary gap in methodological knowledge between academia and practitioners. A closer and more structured interaction can be beneficial both ways, with practice providing realistic problems and case material for methodology development, and academia providing more sophisticated modelling procedures, methods and practices.

Bibliography

[1] Fabien Benureau and Nicolas Rougier. Re-run, repeat, reproduce, reuse, replicate: Transforming code into scientific contributions. *Frontiers in Neuroinformatics*, 11:69, 2018. ISSN 1662-5196. doi: 10.3389/fninf.2017.00069. URL http://arxiv.org/abs/1708.08205.

[2] Jorge Luis Borges. On exactitude in science. borges jl collected fictions (trans. andrew hurley), 1999.

[3] George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356): 791–799, 1976. ISSN 0162-1459. doi: 10.1080/01621459.1976.10480949. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1976.10480949.

[4] Robert J. Lempert. *Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis*. Rand Corporation, 2003. ISBN 978-0-8330-3485-4. Google-Books-ID: F2SzOwE0_bIC.

[5] Ministerie van Economische Zaken. Besluit van de staatssecretaris van economische zaken van 8 november 2008, nr. wjz/8157380, tot vaststelling instructie rijksdienst inzake aanschaf ict-diensten en ict-producten, 2008.

[6] Programmabureau Nederland Open in Verbinding. Actieplan noiv. https://www.pianoo.nl/sites/default/files/documents/documents/actieplannoiv_1.pdf, 2013.

[7] J. Arjan Wardekker, Penny Kloprogge, Arthur C. Petersen, Peter H.M. Janssen, and Jeroen P. van der Sluijs. Guide for uncertainty communication. Technical report, PBL Netherlands Environmental Assesment Agency, 2013.

[8] Mark D. Wilkinson, Michel Dumontier, and et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.18. URL https://www.nature.com/articles/sdata201618.

[9] Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, and Tracy K. Teal. Good enough practices in scientific computing. *arXiv:1609.00037 [cs]*, 2016. URL http://arxiv.org/abs/1609.00037.