

What is a Good Model?

To some extent, building good models is an art. Dijkstra's motto "[Beauty is our business](#)" applies to models as well as to programs. Nevertheless, we can state seven criteria for good models. These criteria are in some sense obvious, and any person with experience in modelling will often try to adhere to them. But surprisingly, our list of criteria has - to the best of our knowledge - not been described elsewhere in the literature, although most of them occur in a [technical report](#) of Mader, Wupper and Boon. (We see this as a clear indication of the lack of interest for the methodology of modeling in our field.) Often, the criteria are hard to meet and typically several of them are conflicting. In practice, a good model is often one which constitutes the best possible compromise, given the current state-of-the-art of tools for modelling and analysis. But a truly beautiful model meets all the criteria! We refer to Mader, Wupper and Boon for further links to related work in the areas of software engineering, requirements analysis, and design.

- A good model has a clearly specified **object of modelling**, that is, it is clear what thing the model describes. The object of modelling can be (a part of) **an existing artefact or physical system**, but it may also be a **document that informally specifies a system or class of systems** (for instance a protocol standard), and it may even be a **collection of ideas of a design team about a system they construct**, expressed orally and/or by some drawings on a whiteboard.
- A good model has a clearly specified **purpose** and (ideally) contributes to the realization of that purpose. Possible purposes include: communication between stake holders, verification of specific properties (safety, liveness, timing,...), analysis and design space exploration, code generation, and test generation. **A model can be descriptive or prescriptive**. If a model has to serve several distinct purposes then often it is better to construct multiple models rather than one.
- A good model is **traceable**: each structural element of a model either (1) corresponds to an aspect of the object of modelling, or (2) encodes some implicit domain knowledge, or (3) encodes some additional assumption. Additional assumptions are for instance required when a protocol standard is incomplete (e.g., it does not specify how to handle certain events in certain cases). Links between the structural elements of the model and the aspects of the object of modelling should be clearly documented. A distinction must always be made between properties of (a component of) a model and assumptions about the behavior of its environment.
- A good model is **truthful**: relevant properties of the model should also carry over to (hold for) the object of modelling. Typically, **for each (relevant) behavior of the object of modelling there should be a corresponding behavior of the model**, and/or for each behavior of the model there should be a corresponding behavior of the artefact. In the construction of models often idealizations or simplifications are necessary in order to allow for the use of a certain modeling formalism or in order to be able to analyze the model. In these cases, the model may not be entirely truthful. The modeller should always be explicit about such idealizations/simplifications, and have an argument why the properties of the idealized model still say something about the artefact. In the case of quantitative models this argument will typically involve some error margin. In the case of nondeterministic models it frequently occurs that a model "overapproximates" reality, and that certain behaviors that are possible in the model are not possible for the artefact.
- A good model is **simple** (but not too simple). Occam's razor is a principle particularly relevant to modelling: among models with roughly equal predictive power, the simplest one is the most desirable. Hence, **the number of states and state variables should be as small as possible, and the level of atomicity of transitions should be as coarse grained as possible** (but not coarser), i.e., **the number of transitions should be minimal given the intended use of the model**. Preferably, things should be written only once, and one should avoid ugly encodings. Preferably, the model uses stable, well-defined and well-understood concepts and semantics.
- A good model is **extensible and reusable**, that is, **it has been designed to evolve and be used beyond its original purpose**. Typically, if one defines models in a modular and parametric way this allows for dimensioning, future extensions and modifications, especially if modules have well-defined interfaces. Ideally, a model should not just describe the specific system at hand: by appropriate instantiation and dimensioning it should be possible to model a whole class of similar systems.
- A good model has been designed and encoded for **interoperability and sharing of semantics**. Model-driven development of an embedded system typically leads to a plethora of models, all presenting different views on and abstractions of the system. If a model is not somehow linked to other models, its usefulness will be limited. Ideally therefore, the relationships between all models should be properly defined, for instance via formal refinement relations.

Clearly, there are many relationships and dependencies between the criteria. If a model is traceable, that is, links between the structural elements of the model and the aspects of the object of modelling are clearly documented, then chances increase that the model will be truthful. Also, if a model has been set up in a modular way, then one may apply a divide-and-conquer strategy both for establishing truthfulness of the model and for analysis. Etc, etc.