

Common Modeling Challenges: A brief guide for aspiring modelers

Hazhir Rahmandad, hazhir@mit.edu

Last edited on: May 30, 2025

Motivation

Learning system dynamics modeling is harder than it first appears. Initial excitement after introduction to the field may be later tempered by the steep learning curve and endless hours trying to get some model to “work.” Existing textbooks and courses primarily offer support for learning the basics of modeling and common formulations. But we lack extensive training resources for when students engage in independent modeling projects. The challenges at this later stage are harder to pin down algorithmically, may call for one-on-one coaching, and students often learn how to deal with them through painful trials and errors.

Some of that pain is inevitable for learning, but a lot of it may not serve any purpose. This document is an attempt to share my personal experiences working with students over the years about some of the common challenges, and possible remedies, relevant for making the learning journey more efficient and rewarding for mid-stage students of modeling. The focus of this document is only on the building of quantitative models. While important, I do not discuss many other aspects such as the development of causal loop diagrams, engagement of stakeholders, incorporating qualitative and quantitative data, and policy design. Moreover, I assume a basic familiarity with dynamic modeling process at the level of the material covered in introductory texts in the field. Therefore, this paper better serves aspiring modelers than those just entering the field.

Even within the quantitative modeling steps there is much that is beyond the scope of this note and covered in a variety of other resources. This document, focusing on the more nebulous aspects of quantitative modeling, is a short complement to that rich literature.

This guide is organized around seven common challenges in quantitative system dynamics modeling and offers partial remedies for each. I start with challenges, both because they are the pain points justifying various recommendations, and because their persistence points to common mistakes that could be avoided. Both the challenges, and the remedies, are somewhat abstract, and perhaps make more sense after somebody has dealt with them in practice. That is why the document is targeting those already familiar with the basics of modeling. In what proceeds I use a direct and informal tone to underline the subjective nature of this content.

Beyond an initial read, this note may be useful as a reference for when you suspect one of the challenges and can help you with ideas for addressing it. For each challenge I try to offer a short definition, discuss the signs for identifying that pitfall early, then discuss various ideas for solving the problem, or avoiding it in the first place.

Challenge 1- Unclear Problem Definition

One of the most common and painful problems of modeling projects is lack of a clear problem definition. The problem definition is the true north of a project that informs every decision, and specifies when a project has reached its objectives. A signal of unclear problem definition is when the modeler can't coherently explain why a mechanism should be in the model and why something else can be left out; why is the chosen level of aggregation fit for the problem; why is the current time horizon appropriate; how much empirical rigor is needed in the project; or when we can call the project complete; among others. Lack of clarity on such basic questions, combined with stakeholder pressures and appeal of "realism" could then drive the project into increasingly complex and incoherent directions, leading to the dreaded "modeling the system" syndrome. Resulting models, even if technically competent at the formulation level, are unwieldy, may collapse under their own weight, and often lack insights needed for real world impact or theory building.

Start with a clear problem definition

The obvious solution to the challenge is to clearly define the problem at the outset. Yet, that definition may take different forms depending on the objectives of a project (e.g. building insight, generating hypotheses, policy analysis, operational decision making, building teaching tools) and evolve over time. Here I briefly discuss three more common types of problem definitions, along with how each may specify a successful model. If you suspect lack of clarity in your project's problem definition, it is a good idea to adopt one of these approaches to problem definition. There are other, less common, options one may consider as well but we focus on this subset in light of their broader relevance.

Explaining behavior over time- Perhaps the most common problem definition type in system dynamics aims at providing a dynamically coherent explanation for a set of observed behavior-over-time dynamics. Typically starting with historical data from a concrete case (or qualitative historical trends), this type of problem definition focuses on finding plausible mechanisms (such as feedback loops, accumulations, decision heuristics) that can generate those reference modes. Key ideas for relevant mechanisms may come from qualitative research, theory, and physics of the problem, providing a natural bridge between qualitative modeling and development of causal loop diagrams (CLDs) with quantitative modeling. Moreover, the mapping between common reference modes (e.g. exponential growth, oscillation) with feedback loop archetypes (e.g. reinforcing loops, balancing loops with delays) assists modelers in the hypothesis generation process and prioritizing mechanisms for quantification.

Extensions of this problem definition type emerge from including in the objective of the project goals such as: providing empirical estimation of competing mechanisms, informing relative strengths of pathways; considering the impact of specific policy levers on future trajectories of the system considering the mechanisms that provide an explanation for the past; and capturing multiple reference modes coming from different cases using the same underlying structure.

Assessing the impact of policy levers- Sometimes clients have a pain area (e.g. weak sales, customer complaints, housing costs in a city), and want to see how a combination of one or a few policy levers (e.g. pricing, investment in capabilities, zoning laws) can be best utilized to address

that pain point. A successful model offers clear guidance on what policy levers to choose at what times to attain a set of objectives that are measurable in the model.

While intuitive for the client, this type of problem definition lacks clear guidance for the modeler on what mechanisms to prioritize in the modeling process. The modeler needs to trace the causal pathways connecting policy levers with outcomes of interest, identify the mechanisms that regulate those outcomes beyond the policy levers at hand, and prioritize the focus of modeling among those mechanisms. For example, zoning laws impact one constraint on housing supply, but other drivers, e.g. housing demand and construction supply, need to also be incorporated if a model is to seriously track the impact of change in zoning laws on housing prices. Absent a reference mode to explain, the potentially relevant components can quickly expand in this example to include businesses, immigration, crime, and so on, risking variants of “modeling the system” syndrome. Complexity of these steps can be reduced by incorporating elements from reference-mode-based problem definition, focusing on a concrete case and the explanation of emergence of the pain point at hand, for example.

This type of problem definition comes with a few implications. First, some level of empirical confidence building may be required, otherwise trading off policy levers becomes mute. Second, assessment of policy levers also often requires models that are robust beyond a single historical case, thus calling for more general formulations and incorporating insights from multiple cases, e.g. using panel datasets. Third, the resulting models may or may not offer clear cut qualitative insights suitable because they were not designed around ‘interesting’ mechanisms. Finally, deeper policy analysis may require beyond simple turning of levers on and off, e.g. using tools from control theory and reinforcement learning. Therefore, this type of problem definition is typically better suited for larger projects with experienced modelers and a budget that covers the empirical grounding and in-depth policy analysis.

Inferring the implications of a set of mechanisms- Models may be designed to illuminate the dynamics that follow a set of structural features and mechanisms. Such problem definitions are most common in building generic theoretical models: e.g. what is the impact of including other-side network effects in a two-sided platform? What happens if we consider the competition between two players rather than assuming a monopolist? What will be the impact of accounting for behavioral responses to risk in epidemic models? Typical models start from a well-known baseline model in the literature and alter a structural assumption, then analyze the implications. More ambitious models attempt to build a new ‘classic’ baseline by incorporating structural features they deem critical in a common real-world problem type. Classical models in system dynamics such as stock management, projects, capability trap, epidemics and diffusion are examples, and many classics in neighboring fields, from Multi-armed Bandit to NK and various seminal economic models also fall into this category. Problem definitions around incremental changes in an existing classic model are promising for research, and more ambitious projects may target the building of a new classic.

These models are successful when they maximize the qualitative insights per unit of complexity: they should be simple but offer clear intuitions useful for thinking about a category of problems. They are not designed to match any single history closely but are expected to inform important aspects of many cases.

Specify the architectural choices of the desired model

The approaches above provide a comprehensive problem definition that then informs various choices during the modeling. A complementary approach is to explicitly make the key architectural choices at the outset and use the choice process to help clarify the problem definition. These choices implicitly depend on the problem definition, and thus their explicit consideration can help clarify the problem definition as well. Four choices are paramount.

Model Boundary and Target Data- Model boundary specifies what is conceptually in and what is out of the model. For example, in a model of epidemics, would you include vaccination? Waning of immunity? Behavioral responses to risk? Demographics of the population? The choice of model boundary is tightly connected with the problem definition, so thinking hard about mechanisms to include, or not, would help you clarify the problem definition. You can start with a larger list of possibilities, typically informed by qualitative modeling and working with stakeholders, experts, and prior theory. Another informative consideration is the explicit listing of patterns in data you want to replicate, which helps focus the outputs you expect your model to generate.

You would then focus on prioritizing those items, perhaps putting them into buckets of essential, important, potential, and marginal. Besides qualitative data, this step would benefit from consulting with quantitative data. Often simple trends, or aggregate statistics, may rule out some mechanisms while giving credence to others. For example, checking the magnitude of hiring during a project life informs whether rookie dynamics may be relevant for the model; or vaccination statistics would be relevant for deciding to include vaccines in the model boundary.

Time horizon- A clear time horizon for the analysis is a simple and powerful part of the problem definition. It will allow you to choose what dynamics matter, and what can be left out. While the maximum length of the horizon is critical, it is also important to consider the minimum horizon of relevance. Dynamics that happen below that minimum resolution can be left out (or simplified into instantaneous impacts), opening the way for more aggregate and less complex models. For example, in a model of population dynamics with the horizon of decades, you may safely leave out weekly and monthly dynamics.

Aggregation Level- Choosing the right level of aggregation is one of the most important architectural choices in a modeling project. For example, populations can be modeled as aggregate stocks, divided by various demographics, or at the individual level, among others. While some choices (e.g. modeling individuals explicitly) may be more intuitive, a hard look at this choice will allow you to avoid excessive detail complexity and trade that off against broader model boundary and more transparent dynamics. Therefore, the general guidance in system dynamics is to choose the most aggregate level which can serve your model's use cases. For example, if you are interested in analyzing racial disparities in health, then breaking down populations by race is inevitable, but including more detailed compartments would require justification. Other reasons for disaggregation may come from the need for more precise simulations. For example, demographic transitions coming from imbalance between youth and elderly can't be modeled with a single population stock, but also do not need disaggregating to the level of single-year cohorts. Finally, full utilization of data in calibration process often benefits from modeling at the level of aggregation that best extracts all of the information available in datasets at hand. This consideration may bias novice modelers towards more disaggregate models. However, adjusting aggregation level for

calibration purposes is a subtle process beyond the scope of the current article and often not the same as choosing the aggregation level in the data. Given the importance of the choice of aggregation level we will come back to this topic under challenge 3.

Sources of inertia- Another key choice relates to picking the sources of inertia relevant to the dynamics at hand, i.e. picking the right stocks for the problem. A helpful practice is to list candidate stocks, spanning those identified from the snapshot test (imagine the system frozen, what is observable is a stock candidate), perception processes (are there significant delays in measurement, perception adjustment, learning or response to information?), and physical accumulations. A systematic scanning of common choices may also offer good candidate sources of inertia for some common problems. For example, in organizational models, sources of inertia are commonly found in: employees (perhaps broken down by experience or skills); customers (broken down by relationship with company (e.g. potential, current, former), product choice, or tastes and features); products (in different stages, from those in design, to raw material to those with customers); tasks; resources (e.g. machinery, intellectual property, money); skills and capabilities; and perceptions (of e.g. quality, productivity).

Once possible sources of inertia are identified, one should attempt at prioritizing these, focusing on stocks with the most relevant time constants (i.e. not too slow- or fast-moving in the time horizon of interest), and key to the dynamics of interest (e.g. in outcomes the client cares about). This advice can become more complex for systems where both fast and slow dynamics matter. For example, a model of electricity capacity planning may have a decade long horizon but also a rather short, even hourly, lower bound on dynamics of interest (due to the impact of peak hourly demand and prices on capacity planning). Problems with these mixed time horizons require careful treatment and sometimes specialized methods.

Finally, a key additional consideration is to avoid conceptual overlaps among the selected stocks. Given the critical importance of picking a good set of stock variables we will revisit this issue under challenge 5.

Challenge 2- Getting Stuck at the Start

Just as the dread of a blank page can delay the work on a manuscript, starting a new model can be intimidating. You may feel lost, not knowing what to focus on first, and doubting your ability to make a difference. A few ideas can help overcome this challenge.

Learn about the problem space

Quantitative modeling should not be the first step of the process, and you need to have done your homework in defining the problem and collecting qualitative and quantitative data to inform the mechanisms relevant to the problem, before you can commence on quantitative modeling. Having built a CLD with the help of stakeholders offers one excellent entry point, though other sources, such as prior theory and field observations can also prepare you for starting to model. When available, bringing in quantitative data into your early exploration of the problem is also very helpful for identifying the key areas of relevance and mechanisms that may matter. Quantitative data also can discipline your model building and analysis process, forcing you to stay close to empirical regularities.

Start simple and add structure in iterations

It is reassuring to remind yourself that “all models are wrong, only some are useful.” Therefore, you don’t need to compete with an unattainable “perfect” model, but can add value with small chunks of structure, as long as those components enhance your thinking (and perhaps that of the client) about the problem. Indeed, an iterative modeling process that adds new components in small chunks, analyzes and understands the resulting model and records its insights, and only then considers the addition of other chunks, provides several benefits. First, it will keep the modeling challenge commensurate with your skills. You can choose the size and complexity of new chunks based on your skill level, and the overall process remains the same. Second, it will allow you to incrementally add the most value you can given the available time. Key is to prioritize more promising directions first. Then you will “be always done”¹, that is, will always have a running model that can (and already has) generated meaningful insights. Third, this approach avoids a common pitfall of building a big messy model which may not be responsive to the problem definition, or may not be easily understood. Fourth, regular checkpoints not only help you, but also enable the stakeholders to remain engaged throughout the modeling process, digesting the incoming insights and building an increasingly rich understanding of the problem. Finally, the iterative approach enables timely adjustment of modeling direction in light of emerging insights and continuous feedback from stakeholders. Mastering this process takes time and practice, but a few pointers are helpful in your early modeling projects.

Start with a single stock- Picking a single stock (or a connected chain of stocks) and formulating its in- and out-flows (with largely exogenous drivers) offers a simple and effective entry point into early modeling iterations. In doing so you want to start with ‘important’ stocks, i.e. those flagged in your discussion of the model boundary and important stock variables. Those choices are often informed by simple trends in data and statistics that allow you to focus on mechanisms most likely to be relevant. Once you feel comfortable with your single stock formulation, you may try to incorporate one feedback loop that may consist of a couple of stocks, or a model chunk that best matches a classical formulation you are familiar with. Remember that a recipe for frustration and failure is to try to model a large set of qualitative loops and mechanisms in one iteration.

Focus on explaining the problem – The north start in the modeling process should be addressing the problem you start with. You want each iteration to get you a step closer to that goal. So you want to continuously assess your model’s ability to explain the problem. For some problems that entails the model’s ability to match some reference mode or quantitative data series.

Analyze and understand before adding anything new- One of the most common failure modes among early modelers is adding structure without fully appreciating its implications. That can create a trap where the modeler, not understanding why the model behaves the way it does, feels unsatisfied with that behavior and goes on to add more structure to align the behavior with his preconceived expectations. Yet, the new structure further complicates understanding and expands the gap between the modeler’s intuition and the model’s complexity. The trap also manifests as changing model parameters and hoping to get “better” results. This is a very time consuming and

¹ Adopted from Prof. Jim Hines lectures in the course 15.875 at MIT

often futile approach because the fundamental problem is lack of understanding and not parameter choices.

The only way out is to fully understand the current model's behavior before adding any new structure. Understanding entails being able to tell a clear (and correct) story of why the model behaves the way it does, connecting structure to behavior. Both analytical and heuristics approaches are available to assist with this understanding, and at their heart is the scientific methods. One needs to come up with hypotheses (in terms of feedback loops, accumulations, exogenous drivers) that explain the model's behavior, and then test those hypotheses by removing the hypothesized driver without affecting other mechanisms (e.g. cutting the hypothesized loop at a unique link). If the behavior changes substantially, the hypothesis is strengthened, otherwise, it can be rejected. Several analytical methods offer automated hypothesis generation (and partial testing).

Good old causal tracing remains a valuable tool for generating those hypotheses. Specifically, you start from a behavior of interest (e.g. population growth turning to decline); then inspect the trajectories of its causes to find the most relevant factor (e.g. a sharp rise in deaths); you then continue to trace back that new 'primary cause' (here the death rate). The process continues until you come back to one of the variables you already had visited (i.e. identify a feedback loop as your hypothesis), or reach an exogenous driver/input into the dynamics (i.e. identify a causal chain with an exogenous driver as your hypothesis).

Another approach to enhance understanding is ruling out unacceptable behavior before it shows up in your model. That requires robustness in every part of the model, from single formulations to larger chunks, so that when complexities add up, you will remain confident about the robustness of larger model and will not need to trace and explain behaviors generated due to bad formulations rather than genuinely interesting structures. A host of testing ideas build on ensuring robustness under extreme inputs and assessing intended rationality (reasonable behavior) under simplified and expected inputs. Resembling 'unit testing' in software development, you can build in robustness in formulating every new equation, and test the broader robustness and logical consistency of the formulation when it interacts with the rest of the model through additional tests. With growing experience, some of the lower-level tests become automatic as you know what the formulation may or may not do, saving you time, but the system level tests remain crucial.

Confirming your understanding- Note that it is very easy to fool oneself into believing a hypothesis and thinking that you have understood your model. Often we see a behavior in the model, come up with a plausible explanation, and accepting that hypothesis with limited effort at falsification incorrectly think that we have understood the behavior. These plausible sounding explanations, when not backed with well-designed tests, fundamentally compromise the value of a modeling exercise and harm the credibility of the modeler.

This problem is so common, even among experienced modelers, that one benefits from testing their understanding in a separate way not dependent on the plausible-sounding explanations. The process is simple: you can predict the model's behavior under various inputs and parameter changes. Would the turn in population happen sooner or later if I increase the population birth rate? After writing down the predictions you conduct those simulations. Some software programs simplify this process, for example with a single click Vensim allows you to change all model

parameters by a given percentage and track some objective function. If your predictions are uniformly correct you can declare victory, otherwise you need to continue with the analysis step.

Build testing into models: equilibrium and infrastructure- The testing of alternative hypotheses, and your intuition for model's dynamics, will benefit significantly from two practices: putting models in equilibrium, and adding structures needed for systematic testing and tracking. Starting from equilibrium allows you to introduce controlled experiments (e.g. step or pulse inputs) without transient dynamics confounding your understanding. A model is in equilibrium when (expected) net flows for all stocks are zero. Some models lack useful equilibriums (e.g. a project model is in equilibrium when no work is done, or the project is finished), but for others one should seek equilibrium before effective analysis can be conducted. Solving for equilibrium analytically is preferred, but not always feasible, in which case you can put as many model stocks as possible in equilibrium, and find the remaining using numerical or optimization methods.

The second suggestion for facilitating analysis is to build the machinery that simplifies testing and intuition building. You can start simple: e.g. add a view with key parameters, switches, and graphs of outcomes; introduce explicit switches for cutting loops or separating different sectors of the model; include exogenous input functions (such as step, pulse, oscillation, and random). More comprehensive methods will pay off in more complex projects: defining a suite of tests automated through software (e.g. reality check feature in Vensim, running multiple tests in a script, using SDEverywhere features, or conducting tests using external code and DLL connectivity); utilizing analytical tools (e.g. Loops that Matter in Stella); or using optimization to try to break a model (e.g. changing parameters in feasible ranges to get impossible outcomes).

Challenge 3- Poor Aggregation Choices

Aggregation is inevitable: even for simple-looking concepts, such as populations, a choice should be made between options ranging from subatomic particles all the way to cells, organs, individuals, or aggregate population groups. A regular challenge to quality models and a robust iterative modeling process is picking a bad, or inconsistent, level of aggregation. Too much detail and you will be lost in your model's detailed complexity, spending a lot of time to get minor things "right" while missing the more consequential mechanisms. Too much detail can bug one down in searching for irrelevant data, grounding a proliferating number of unimportant parameters empirically, or getting less consequential structures right. Models could also become too aggregate when they aggregate away mechanisms key to problematic behavior modes, miss to capture relevant policy levers, or fail to connect the model with data available for estimation. Another common problem is when different parts of a model have incompatible levels of aggregation. This leads to awkward interfaces attempting to connect those sectors, and also wastes resources. For example, details that may explain 1% of empirical variance in one sector may consume a large share of effort, whereas another sector is modeled so aggregate that it injects an order of magnitude more uncertainty into the overall model dynamics. Level of aggregation is thus one of the most consequential choices in a modeling project, and one for which few generic rules exist. It is not surprising that many novice modelers find it challenging. In fact, it is common that the

subtleties in this choice hide its critical role leading some modelers to make it automatically and with little thought.

Carefully consider aggregation choice early on

While there is no simple rule for selecting the aggregation level, considering the key tradeoffs can help with making more informed and conscious choices. You need to decide which of these considerations are paramount for your specific problem, and adjust your level of aggregation accordingly.

Benefits of Aggregation- Starting aggregate can help one build early versions of the model more rapidly, iterate, and revise based on client feedback and emerging insights. Side-stepping details also can help with communicating with clients and building generic insights from the outset. Aggregate models have fewer structures and parameters, simplifying model analysis, testing, and estimation. Reduced computational cost can further expand the reach of sensitivity analysis and formal parameter estimation. Aggregate models are usually better received in academic outlets, especially those interested in theory. Thus aggregate models allow for more cumulative modeling when others build on your work and offer more effective carriers of insights to different audiences. Building quality aggregate formulations is however not easy, and requires subtle compromises which may require significant experience.

Benefits of Disaggregation- Disaggregation offers its own advantages. First, it is easier to stay more operational (i.e. connecting formulations with visible and intuitive processes) in more disaggregate models. Relatedly, existing theory often has much more depth in a narrow part of the model, better supporting disaggregate models and their face validity in the eye of other experts. Disaggregation enables more precise treatment of different components and their interactions, promising more internal consistency and precision in model outputs. In fact, sometimes disaggregation is needed for discovering emergent mechanisms that may not surface in aggregate models. For example, the impact of network structure on diffusion dynamics can best be revealed in models that disaggregate to agents and model their communication networks. Disaggregation also enhances the ability to capture relevant policy levers, improving face-validity and communication with stakeholders interested in those levers. Finally, disaggregation offers closer connection between the model and the relevant data when detailed measurement is available.

Enforce consistency- While the considerations above point to the importance of picking the aggregation choice based on the requirements of the problem at hand, a consistent level of aggregation across different model chunks remains helpful regardless of the chosen level. That is facilitated by the iterative modeling process which adds new components in light of what is already captured, and could use the level of aggregation in previous components as a guide for upcoming ones.

Thinking about the interfaces between different chunks of the model (e.g. using price signals to connect supply and demand sectors) and ensuring consistency across those interfaces is another useful guidepost. Clean interfaces also enhance modularity in a model, enabling faster testing of different modules, partial model calibration, collaborative modeling (where different people can focus on different modules once interfaces are agreed upon), and enhancing efficiency by using existing formulations as sub-modules.

Be flexible- Modeling communities often have historically-preferred choices on aggregation, which may bias their assessment of this choice. For example, defining oneself as an “agent based modeler” may force one to choose an individual level of aggregation, whereas a “classical system dynamicist” may exclude individual level models from consideration set. Each may be suboptimal for some problems, so flexibility and explicit aggregation choice are critical for building useful models. Flexibility may also be important because iterative modeling can lock one’s model into an aggregation choice made earlier. Not only it is important to consider that early choice with an eye on future expansion of the model, but also one needs to be open to serious revisions in a model’s structure and level of aggregation when evidence accumulates in favor of better alternatives.

Challenge 4- Overlapping and Unclear Constructs

Models that lack clarity in their variables can’t go far and may confuse users rather than offer insights. For example, consider a model of employee performance that incorporates distinct variables for morale, burnout, commitment, and motivation, with separate impact of each on productivity. The effects may sound plausible, but are likely double counting the same underlying mechanisms, and building each variable on similar antecedents, creating a confusing web of overlapping concepts and causal relationships. This common problem often manifests in two ways: variables that are vague in their definition, and ones that are overlapping in their conception with other model variables. This problem is sometimes exacerbated when models are aggregate (and therefore variables are more encompassing and less concrete). Regardless of aggregation choice, a few guidelines are helpful to minimize the risk.

Stay close to a case

Concrete cases, from a company’s history to a country’s infection trajectory, simplify the grounding of model variables in case tangibles, and thus more clear definitions. Therefore starting with a concrete case in mind is good practice, even if you are targeting a generic model that goes beyond your single case.

Define variables clearly and avoid overlap

It may sound trivial, but a clear variable definition is indispensable. Make sure you think through what is included in each variable, and what is not, then record your definition and its boundaries in the comments section for variable in your simulation software. When you define another variable in the conceptual vicinity of the first, ask yourself if both are needed, and how different they are. A good heuristic is to follow the MECE (Mutually Exclusive and Collectively Exhaustive) principle in modeling. This applies rather strictly to the definition of stock variables: outside of carefully considered exceptions they should not have overlapping members. For example, having two stocks for *children*, and *total population*, is bad practice because the two overlap; instead you want stocks for *children* and *adults*, with an auxiliary variable that calculates total population by summing those constituents. The MECE principle is useful more broadly: even auxiliary variables should avoid significant conceptual overlap, unless they are used to create each other, as in *total population* being calculated based on *children*.

Challenge 5- Poor Stock Selection

Stocks (state variables) are at the heart of a model's dynamics. Picking the right stocks allow you to model the dynamics of interest with minimum complexity. Moreover, structuring iterative modeling around model chunks focused on different stocks is often a promising process. Yet the selection of the best stocks does not follow any simple rules, and becomes a common challenge for early modelers. The problem can be better addressed if it is broken down into two distinct steps we discussed above: the selection of key inertial concepts, and picking the right level of aggregation within each concept area.

Once an inertial concept and a level of aggregation is chosen, the MECE principle provides a helpful heuristic for organizing stock and flow chains or state transition maps for each inertial concept. In building those chains careful attention to time constants will help keep the model in a coherent level of aggregation. Specifically, stocks that move very fast (compared to the time horizon of interest), could be aggregated together, or approximated as auxiliary variables. For example, in modeling decade-long human population dynamics, mosquito population (e.g. relevant for Malaria morbidity and mortality) may be modeled as an auxiliary variable dependent on environmental factors. On the other hand, stocks that move too slowly may be assumed as parameters. For example, typical models of acute epidemics assume a constant total population and ignore demographic dynamics.

Another helpful heuristic in stock selection is to model the most fundamental source of inertia and let that inertia reverberate in the rest of the model. For example, in modeling human body weight dynamics, total weight is a clear 'stock' but also depends on lower level stocks such as 'fat mass' and 'fat-free mass'. Modeling the latter as stocks (and formulating total weight as their function) is consistent with MECE principle and avoids confusion and double-counting of inertia.

Finally, having a prioritized list of stocks (discussed above) provides you with a helpful roadmap to prioritize model chunks to be iteratively added. Therefore, start with creating a large list of possible sources of inertia, but prioritize them, and only spend time on mapping their components or detailed formulations once you have decided that one should be the focus of your upcoming iteration.

Challenge 6- Bad Formulations

If you have a clear problem definition that has informed appropriate level of aggregation and concepts, and you have chosen a good set of stock variables, you already have significantly simplified your model formulation and have avoided many failure modes. Yet you still can make poor formulation choices. Modeling practices that lead to bad formulations have many commonalities, allowing for process tips that could enhance formulations significantly.

Learn and use the standard formulations

Over the decades modelers have developed many standard formulations that cover most of the common modeling problems you may encounter. These formulations are well-known (and thus simpler to explain), robust, and versatile, significantly reducing the complexity of model building. In

fact, early on, it is a bad sign if you attempt to come up with a novel formulation: you are likely trying to re-invent the wheel, and the result is likely to be a poorly designed wheel. So always start with standard formulations, and only innovate when you are confident no existing alternative works. Genuine new formulations are rare and deserve their own publication.

Repositories of standard formulations- Some domains include explicit repositories for these formulations. For example, ‘Molecules’ for system dynamics² offer more than two dozen common structures that may well cover 80% of formulation issues a novice modeler would face. Learning those is key to building modeling skills. That learning entails three components: you need to know the structure and equations, you need to know its use cases, and you need to know the behaviors that structure may generate, as well as those it can’t generate. Study of molecules one by one would be helpful but for internalizing the key insights you need to apply them to concrete problems, and go through the analysis and testing of their behavior with different inputs (see the analysis process discussed under Challenge 2).

Replication and textbooks- Besides such repositories, modelers can learn those standard formulations through replicating prior quality models in the literature, making replication a fruitful path to building modeling skills. Typical system dynamics textbooks also go through these standard structures, often dedicating whole chapters to one or a small number of formulation chunks such as *diffusion*, *stock management*, *projects*, *capability traps*, *aging chains*, and others. Many of these more complex building blocks are created using simpler components such as *proportional split*, *producing*, *delay*, *co-flow*, etc.

With this knowledge you can quickly scan your mental repository of existing formulations for one that fits a new problem setting, and implement a suitable alternative efficiently. Expert modelers typically look at a modeling problem and mentally identify the formulations that would fit the different parts of the model. This process of ‘seeing the molecules’ inside a problem will offer you a clear roadmap for your iterative modeling journey from the outset.

Build robustness by explicit requirements

While the physics of the problem offer helpful guidance on how new formulations can be designed (to approximate the structural features of the problem), that guidance only goes so far and may not apply for more abstract models. Thus, in putting together formulation chunks it is critical to be explicit about the requirements and use them to screen and design appropriate formulations: What are the intended objectives of the chunk, and does a proposed formulation achieve those objectives? What are the inputs and outputs for the formulation? What ranges of inputs are acceptable, and are those enforced by the other components that provide input to the focal chunk? What are the outputs needed from the formulation? What are the acceptable ranges for those outputs? How can you enforce those output ranges in the focal formulations? What are the well-established regularities between inputs and outputs that need to be reflected? What extreme

² Dr. Jim Hines has collected a set of established structures under the name ‘molecules’ and connected them in a hierarchical structure available at <https://sdmolecules.org/index.htm> as well as in some of the simulation environments under the name Molecules (in Vensim) or Assemblies (in Stella).

condition tests can inform the robustness of the formulation? Does the formulation offer the flexibility to adjust the strengths of key mechanisms through intuitive parameters?

Addressing all these questions explicitly may feel like a big investment; yet early on the exercise pays off in time it saves avoiding bad formulations which both compromise the model functionality requiring major revisions, and extended analysis time to identify those failures. As you build intuition for classical structures, these considerations become second nature to your modeling and allow you to more efficiently choose the right structures and intuit their local robustness and fit without explicit testing. This approach will allow you to build robustness into every equation and chunk of the model, significantly reducing the risk of system-level robustness failures down the road.

Other suggestions for enhancing formulations

Beyond following standard formulations and building robustness through explicit requirements, a few heuristics are generally helpful and not too detailed, so I briefly discuss them here.

Limit inputs of formulations- Single equations in your model should be easy to understand, so that you can think through their implications, ensure robustness, and communicate their logic to others. That often requires limiting the number of inputs into an equation, perhaps to three or fewer, so that the equation has only one major role rather than trying to accommodate multiple goals in a single formulation. This suggestion may lead to a few more auxiliary variables, but the gains in clarity and enforcing robustness outweigh the costs. Moreover, by spelling out the different steps of a more complex logic you can inspect each step during analysis and identify what part of the formulation is responsible for the observed behaviors, or bugs. As you build more modeling skills you may occasionally violate this ‘rule of 3’ with careful consideration, but the benefits in analysis and communication with others remain regardless of your modeling skills.

Avoid discrete formulations- This suggestion is rooted in the continuous nature of classical system dynamics models. In these models discrete formulations, such as ‘IF’, ‘MAX’, ‘MIN’, tend to be used by novice modelers for two problematic purposes. First, some use them to enforce extreme condition robustness without carefully considering the underlying mechanism that in the real world enforces the robustness. Second, these functions enable building models that rely on discrete thinking (about time and activities) in a context where discrete thinking can compromise model quality. Thus early on it is a good idea to avoid these equations except when they are really necessary.

Ensure unit consistency- While unit consistency does not guarantee a good formulation, any good formulation should have consistent units. Inconsistent units often point to fundamental conceptual problems. So use unit consistency as a helpful guide and enforce it at every step of the modeling (rather than adding units after adding equations).

Challenge 7- Chasing a Desired Behavior

We briefly discussed this failure mode in explaining the iterative modeling process; but it is so common and such a sink of time that it merits explicit highlighting. This primary failure mode in model analysis follows a regular script: the model is simulated, the modeler finds the behavior problematic, so s/he comes up with a guess about the source of the problem (often a mechanism

being too weak or too strong), and tries to fix that by changing the parameters that control the mechanism's strength. In most cases that does not solve the problem, so the cycle is repeated many times, changing multiple parameters, even structures, sometimes in combinations, often with limited success. Occasionally the fix seems to solve the problem, so the modeler stops the analysis, thinking that the model is now acceptable. Note that this process dooms the project regardless of the outcome. A long search for the desired behavior takes a lot of time and often leads to frustration but little learning. Even if some parameter combination allows the model to behave the way the modeler expected, there has been no structural insight. All the model has done is to show a behavior that is consistent with the initial qualitative expectation from the modeler. Yet, absent careful analysis (discussed under Challenge 2) the chances are that the modeler's espoused theory of why the model behaves as observed is wrong; most those theories are! To make things worse, the successful search for getting the expected behavior blinds the modeler to the possibility that their understanding is wrong, and allows them to declare success when the model has added little insight.

As discussed earlier, starting simple and adding complexity iteratively reduces the risk of getting to a very complex model that you can't analyze and understand. Within each iteration the solution to this problem is twofold. First, chasing a desired behavior is a costly trap with little benefits; just avoid it. Second, focus your analysis on understanding what structures in the model generate its current behaviors. That understanding comes from following the scientific method by going through cycles of hypothesis generation and hypothesis testing. Your focus should be on refuting your hypotheses and not on chasing a desired behavior. For many pedagogical and theoretical applications the resulting understanding is the primary fruit of working with a model: that is how you build new structural insights that allow you and your audience to think more clearly about the problem. Those benefits do not depend on matching some expected behavior. Once you have understood the exact structures generating the current behavior (expected or not) you may decide to revisit some of the parametric assumptions, if you have new insights that suggest initial baseline values are problematic. More often you learn that your expectations were incorrect, either because you did not mentally account for some existing mechanism, or you have not yet included mechanisms that matter in the model, but your mind is accounting for them. Both offer useful insights, make sure you write those insights down before you get used to them and forget their novelty, and before you start any new iteration by adding new structures.

Final Thoughts

Modeling is as much an art as it is a science. The craft is imbued with subtle judgment calls from architectural choices on time horizon, level of aggregation, and stock selection to robust and flexible formulations. Mastery of this art promises deep joy and an appreciation of beauty in turning complex problems into understandable, sometimes counter-intuitive, dynamic stories. Yet the flip side of the ambiguity in the craft is an arduous journey of learning through trial and error. Many aspiring modelers may decide they cannot afford the investment needed to get there. Yet the journey can be much more pleasant if one learns about, and avoids, the common challenges. I hope this article provides some structure to detect those challenges early on and respond along more promising pathways.

MIT OpenCourseWare
<https://ocw.mit.edu/>

15.879 Building Serious System Dynamics Models from Scratch
Spring 2025

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.