

Deeper Dive into Static Equilibrium and Dynamic Modeling

Using computational models to represent complex systems can greatly enhance our understanding of how such systems work. The modeler can test their understanding of a system's behavior by sketching out its structure, defining the connections within it, and simulating to see if known system inputs will result in outputs that match the real world. By comparing a model's output to observations from the real world, the model can be iteratively revised, advancing toward a more accurate representation of reality. It is the process of evolving one's conceptual model in tandem with the gradual development of an accurate computer model that constitutes learning and makes the process of modeling so powerful in the classroom.

SageModeler offers two distinct approaches to computational modeling, static equilibrium and dynamic time-based modeling. The modeling type you choose will depend upon the purpose of the model.

What's the difference between static equilibrium and dynamic time-based models?

Static equilibrium models

Static equilibrium models are used to represent systems that are inherently stable, or can be simplified such that the state of the system is defined by the combination of inputs to that system. Let's use a simplified example of an epidemic.

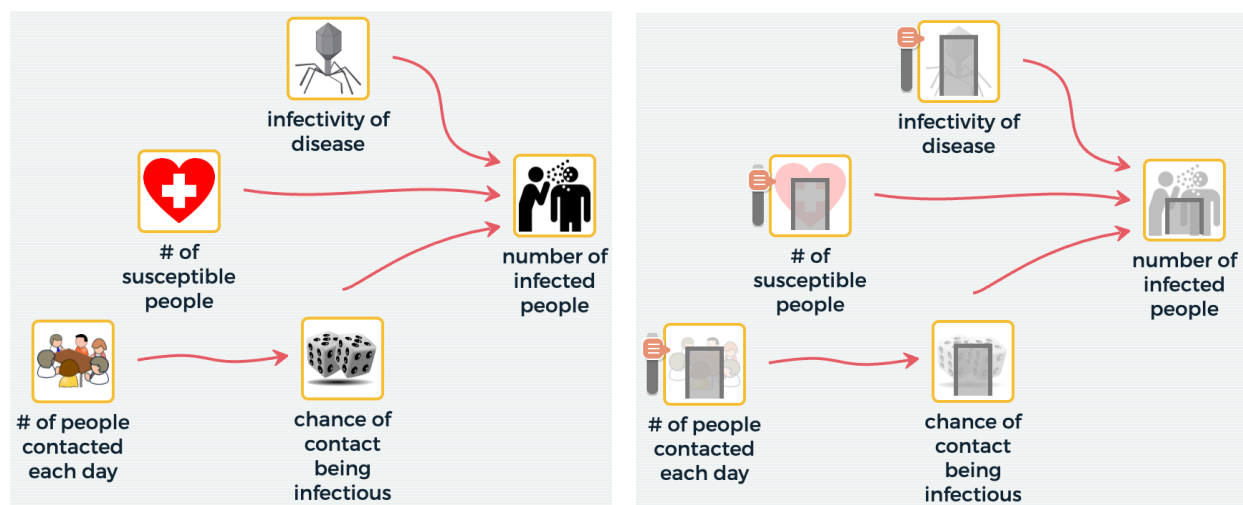


Figure 1. Modeling the outcome of an epidemic (left). During simulation, bar graphs represent the value of each variable (right).

Imagine that you are interested in creating a model to help you determine how bad an epidemic will be based on several factors. A static equilibrium approach would first identify the critical

elements of the system affecting the number of people who will get infected and then define how each acts in combination with the others (Figure 1, left). The combination of susceptible people, infectivity of the virus, and infectious contacts will behave somewhat like an algebraic equation, where a change to one variable causes all other variables to respond as dictated by the relationship rules among them. With each change of a variable, the entire system immediately adjusts to a new static and stable state or outcome (Figure 1, right).

In the epidemic model above, when *# of people contacted* is increased, a new state of the system is determined by combining this new input value with the other variables in the system, increasing the *number of infected people*. Each system “state” results from a particular set of relationships among known system components, and an adjustment to any of them will cause the system to instantaneously reach a new static equilibrium. It is as if you took a photograph of the population at one point in time, increased the number of contacts and took another photograph, then compared the pictures to analyze the effect of *# of people contacted each day* on *number of infected people*.

In a classroom, static equilibrium models provide an excellent way for students to express their understanding of structural complexity in systems with multiple components and interconnections. For many topics it can be extremely powerful to have students move from a list of important components to a diagram (on paper or a model diagram in SageModeler) that shows how the components are causally interconnected, and finally to a computational model that defines the rules governing the relationships between variables. What the computer offers is the ability to simulate all of the connections simultaneously and provide output(s) showing resultant changes across the entire model. Static equilibrium models can be simple, as in the epidemic example, or can be built out indefinitely, to the level of structural complexity desired by the teacher or the student.

Structural complexity can be **deepened** by including detail about any of the system variables, or **broadened** by adding variables from outside the model’s boundary. In either case, a model can become quite complex. It is up to the teacher or modeler to consider an appropriate boundary for the system being modeled.

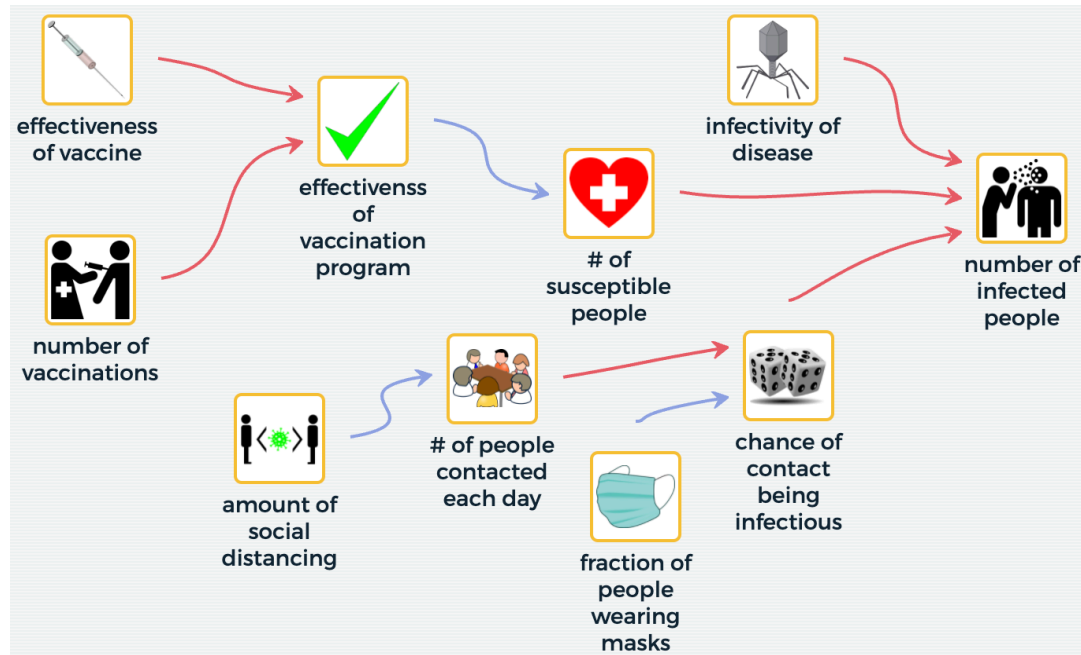


Figure 2. Broadening the epidemic model to include mitigating factors like social distancing, masking, and a vaccination program.

In the [sample model](#) shown here (Figure 2), the original model boundaries have been expanded to include variables representing mitigating factors for controlling the epidemic. In a similar manner, a student might deepen their representation of how the virus infects and reproduces at the molecular level, or expand the boundaries further to include economic and social impacts.

It is the ability to represent one's growing understanding of the many components in a complex system combined with the capacity to reliably simulate and test the validity of that understanding against reality that makes static equilibrium modeling so powerful as a learning tool.

Dynamic time-based models

Static equilibrium modeling can be a powerful tool for expressing and assessing one's understanding of the scope and interconnectedness of a complex system, but it lacks the ability to address a system's potential to evolve over time. When chains of causal connections loop back upon themselves, like a dog chasing its tail, the state of a system at one moment in time can provide the impetus for change into the next, and moment after moment the system gradually evolves. Most of the systems we encounter each day behave in this manner, and dynamic time-based modeling is the more appropriate approach for representing them.

Let's revisit the epidemic example. From a dynamic perspective it can become a story of change over time. Rather than modeling the overall severity of the epidemic, we may want to model how many people will get sick over time. This was particularly important during the COVID-19 epidemic in order to plan for expanding hospital capacity or when to ease lockdown conditions.

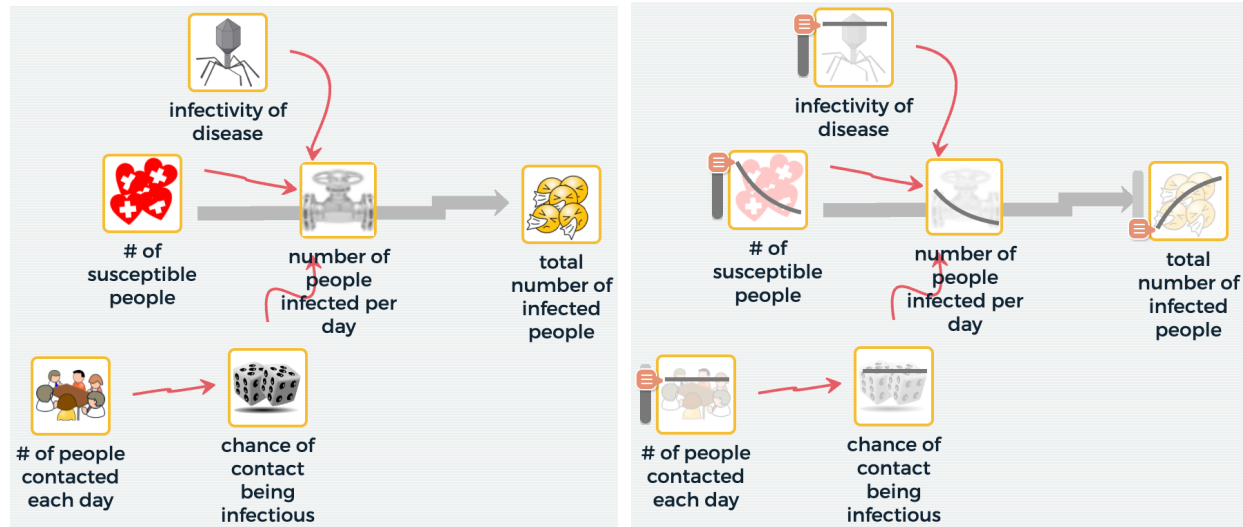


Figure 3. Dynamic time-based model of a collection of susceptible people becoming infected over time (left). During simulation line graphs show the values of each variable over time as the model runs (right).

To see an evolution of the model state over time, dynamic models include new kinds of variables called *collectors*, which can be added to or subtracted from during each model calculation cycle, and *flows*, which define the rate of change in the collectors (how much is added or subtracted each cycle to the collectors). In Figure 3, the *# of susceptible people* is a collector, as is the *total number of infected people* variable. The *number of people infected per day* variable represents how many people become sick each cycle of the model, causing that number to be subtracted from the *# of susceptible people* and added to the *total number of infected people*. This flow, represented as a picture of a valve, controls the rate of transfer from susceptible to infected collectors. Other variables, like *chance of contact being infectious*, *infectivity of disease*, and *# of susceptible people*, all affect the *number of daily infections*. Over time, the *# of susceptible people* decreases and the *total number of infected people* increases. However, we can see the *number of daily infections* starts high and slows down over time, due to the fact that there are fewer and fewer susceptible people over time to get infected.

Because collectors can only be added to or subtracted from, they have a kind of “memory” of the state of the system. In their role as the model’s memory, collectors make it possible for dynamic time-based models to show how feedback causes the system to influence itself. In a feedback loop, the computer first ascertains the model “state” at a given moment by noting the values of all the collectors. These values influence other system variables based on relationship rules designated by the modeler, and eventually, through a chain of cause and effect relationships, loop back to influence a variable that is directly connected to the collector that initiated the loop. In Figure 4, an additional relationship has been defined between the *total number of infected people* and *chance of contact being infectious*. This then connects to *number of people infected per day* and back to *total number of infected people*.

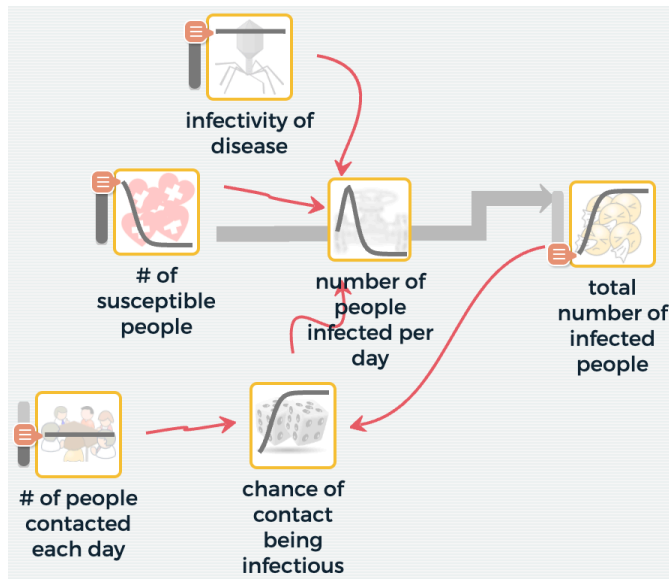


Figure 4. Feedback added to the system by connecting the *total number of infected people* to the *chance of contact being infectious* variable, completing a loop with the *number of people infected per day*.

Notice how the *number of people infected per day* more correctly shows the pattern we observed in the real world, when infections start slowly, reach a peak, and then taper off. Most complex systems include feedback, and adding a feedback loop into our model causes it to more realistically model the phenomenon of an epidemic.

Let's look at another example (Figure 5) with two feedback loops and walk through the logic of how this system will evolve over time.

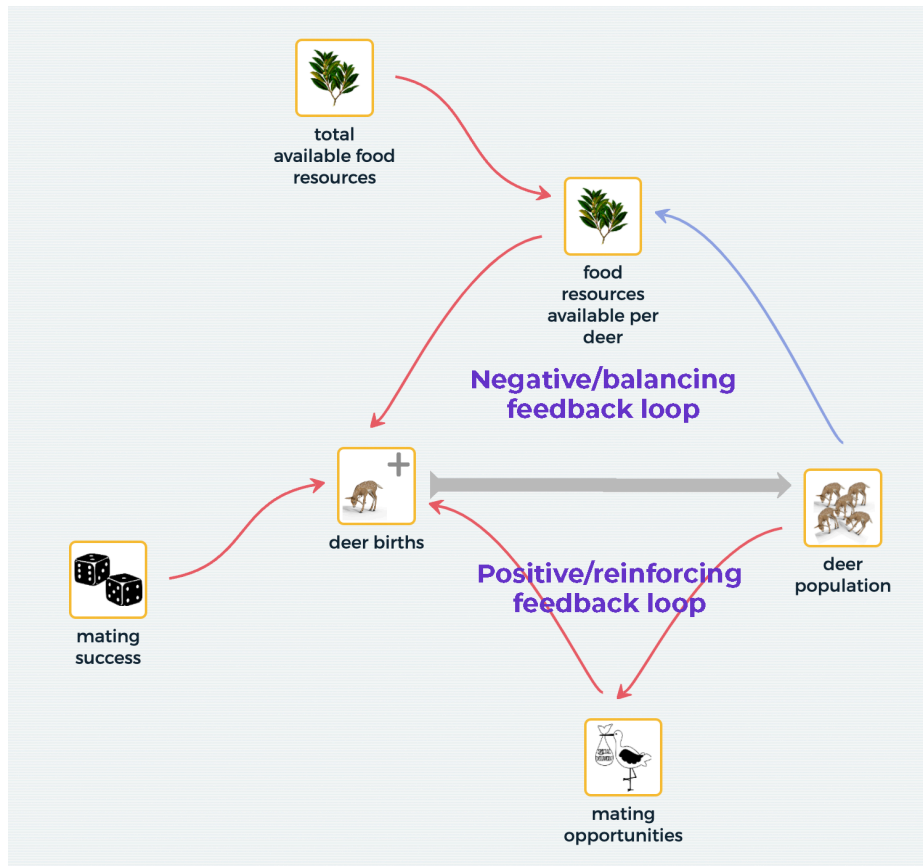


Figure 5. Two feedback loops affecting the *deer population*, the collector.

When calculating how the model will behave from moment to moment, the computer begins with the collector— *deer population*. Tracing the “Positive/reinforcing feedback loop,” the number of deer in the population directly influences the number of *mating opportunities*, which together with *mating success* results in new *deer births* that are added to the population. The connection of this final variable back to the collector completes the feedback loop. In this case, the presence of deer in the population will feed back to add more deer to an already present population. If this loop described the entire system, then the deer population would grow exponentially with more deer having more births, causing more deer in the population and so on.

However, there is a second loop in this system, one that defines a “Negative/balancing feedback.” In this part of the system the *deer population* is connected to *food resources available per deer* in such a way that an increase in *deer population* will cause a decrease in *food resources per deer*, which also impacts the *deer births*, causing them to decrease as well. Fewer *deer births* causes the *deer population* to grow more slowly, balancing out the system so that it can’t grow unchecked.

What type of model should be used?

With a more robust understanding of the differences between static equilibrium and dynamic time-based modeling comes the question of when it is best to use each approach. Dynamic time-based models are most powerful when dealing with fundamental questions about change in a

system's behavior as time passes. Exploration of feedback-induced behavioral patterns like exponential growth, growth (or decline) to a limit, S-shaped growth, and oscillations are the sweet spots of dynamic time-based models. Static equilibrium modes are at their best when exploring the breadth and complexity of connections among a multitude of variables in a system.

Before deciding on the modeling approach, it is critical to clarify the model's purpose, and to do this often requires careful consideration of the question you wish the model to answer or the phenomenon you wish it to represent. Ask yourself, "What do I want my students to learn by building this model?" If the priority is analysis of the structure and interconnection among components in a system and how a change to a system input will affect other system components, static equilibrium modeling is probably best. Static equilibrium models are excellent when asking a question like, "Within the system I have modeled, if a change is made to x, what will happen to y?"

On the other hand, if the priority is to investigate why a particular behavior over time is seen in a system, or how a change to some variable will alter the way a system develops over time, a more appropriate approach will be the use of dynamic time-based modeling.

The bottom line is that different questions require different modeling approaches. Here are some examples:

Q: If CO₂ production from factories were cut to half of what it is today, how would that affect global temperatures?

Model type: Static Equilibrium ([example 1](#), [example 2](#))

Q: If CO₂ production from factories were cut to half of what it is today and allowed to continue at that level into the future, how would global temperatures change over time?

Model type: Dynamic time-based ([example 1](#), [example 2](#))

Q: What are the important components responsible for respiration within a cell and how are they connected?

Model type: Static Equilibrium ([example](#))

Q: Why is it that cells do not continually increase energy production even when supplied with unlimited raw materials?

Model type: Dynamic time-based ([example](#))

Summary

Building computational models can be extremely powerful in a classroom. Modeling offers students the ability to test their understanding of the system as a whole. Static equilibrium models are best used when examining the structure of a system and questions of how a change in one part of a system can cause change in another. Their behavior is extremely stable, and very large systems can be explored without fear of encountering much confounding behavior.

Dynamic time-based models are best when addressing questions of fundamental patterns of behavior over time. Through the construction of simple models containing one or two collectors and a few feedback loops, students can gain appreciation for the importance of circular connection (feedback) and accumulation in complex systems, and how the state of a system evolves over time.