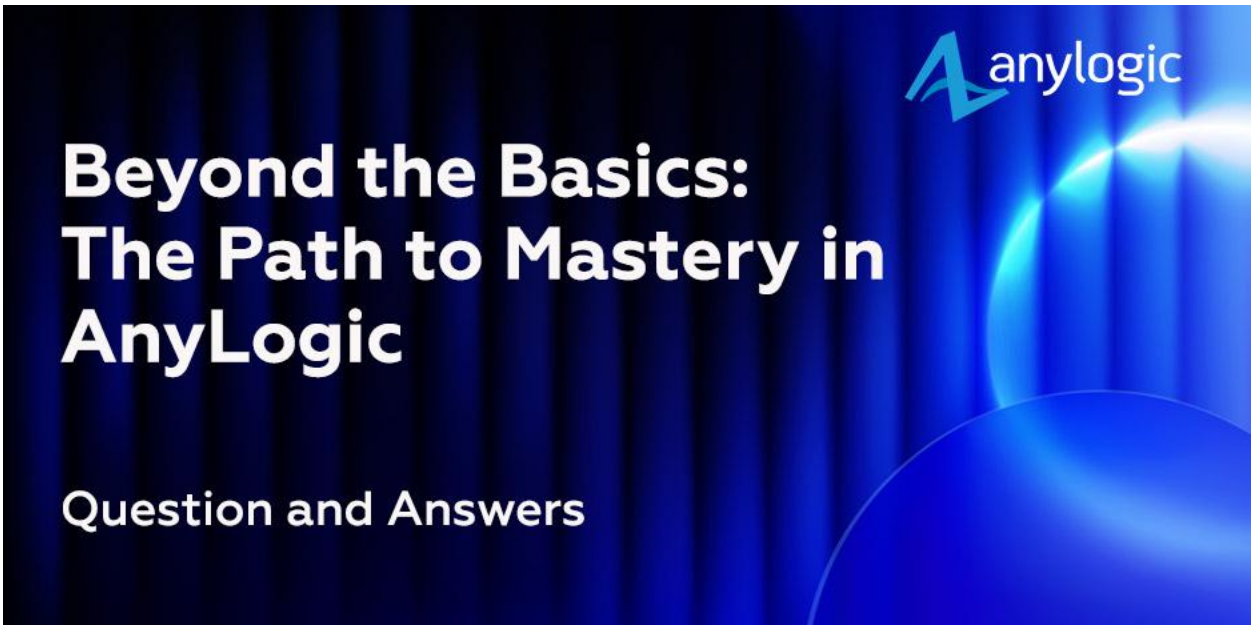


“If someone tells you something like “AI will do all that for you” or “No coding required,” they are liars.”

Andrei Borshchev

“Build, build, build. Only training makes you a master.”

Benjamin Schumann

“You create a model to make better decisions and navigate (hopefully better) towards a still unknown future.”

Stefan Bengtsson

Question Sections

AI

Q 1. Do you see GenAI having a role in augmenting simulation models? (Mike from the United States)

Andrei:

Absolutely.

During design time:

1. Help with conceptualization for non-experts.
2. Assistance with writing script snippets in the model/putting together fragments of model constructs.
3. Explaining errors and tracking down error sources (such as compilation errors or runtime errors).
4. Experiment design.

Output analysis:

1. Interpreting simulation results, finding causalities, and connecting the dots.

Ben:

I hope that simulation tool providers will train their own GenAI LLMs such that they can actually produce base models or adjust existing models. So, I can ask, “Create a simple warehouse with these details,” and I only need to adjust this instead of starting from scratch.

Q 2. Will AI-integrated tools (such as those powered by OpenAI's GPT) soon revolutionize simulation modeling by automating model creation, optimizing parameters, and enabling real-time adaptability, ultimately diminishing the necessity of learning traditional modeling processes on platforms like AnyLogic through tutorials? (Avichal from the United States)

Andrei:

No. The process of model creation will stay human-driven, and humans will be responsible for making the most important decisions: choosing the exact detail level (= making assumptions), validating the model, working with SMEs (subject matter experts), and debugging the model when it behaves not as expected. If you don't know the modeling language, you will be unable to accomplish those tasks.

Ben:

Creating models: somewhat. Optimizing: no. But you will still need to learn how to model. Most training data for great models is inherently secret, and LLMs can only produce what is accessible, which is mostly trivial or bad models.

Stefan:

Answering questions 1 and 2.

Probably given the potential. The technical possibilities to realize this are there (and will most likely improve further). That is not saying that this by necessity is a good thing, though, and I think **it would be wise to move in this direction with care**.

A comparison and analogy can be made with the technical possibility to "model without programming." Yes, it will most likely make it possible to handle some of the simple modeling challenges faster and make simulation more accessible to a larger group of people (and it allows you to visualize the logic, if the platform is designed that way—like AnyLogic). But to think this in a good way can give the possibility to create models with the necessary level of quality to add value (given real-world challenges), which is usually naïve. You need the ability to tailor-make the model, adapting it to the specific real-world challenges, through sophisticated coding options. In AnyLogic this is handled in a good

way, by allowing the modeler to add code related to more or less every logical building block ('On entry,' 'On exit,' etc.).

So, if AI is introduced to speed up the modeling, I **claim there still must be elaborate parallel possibilities to "not use AI" and to "remodel" whatever the AI process suggests**. Another danger with letting AI take over the modeling phase is that over time it **will create less competent modelers** (who in the end will have a hard time questioning the results of the AI process, which would be risky). And not to forget, you would lose **the competence-developing aspect of the modeling process** as such. Modeling is one of the best ways to learn about processes, systems, production, decision-making, or management thinking in general (probably especially operations management). If you automate this, you will lose this aspect—the fact that modeling develops the individual (the modeler), which is competence the individual can apply in "the real world" (also when not dealing with modeling and simulation). So sometimes **the road to get to the end result can be as important as the end result**!

Model reusability in AnyLogic

Q 3. What do you think of model reusability in AnyLogic? Most of the models are problem-specific and dead after serving the purpose. What do you think about having reusable models? (Eyup from the United Kingdom)

Andrei:

Good question. First, these days more and more models are used operationally, so they are not dead after use. And we (AnyLogic) are doing everything to ensure a longer and healthier lifecycle of simulation models (AnyLogic Private Cloud—the execution environment). But your question was, I guess, about reusing models from one project to another or reusing model components.

As to the latter, libraries in AnyLogic are examples of reusable components (as you know, library objects are essentially AnyLogic agents created in AnyLogic). Large organizations (consulting and industry) create their own in-house libraries. Some make them public. Some sell them.

As to reusing whole models, that only happens within organizations that have identical or close to identical facilities, sites, and business units. Otherwise, specific modeling goals and specific assumptions made during model creation (and there are a lot all the time) make reuse of larger models virtually impossible.

And a side note: documentation that should be provided with reusable simulation models or components is more challenging to write compared to software components; in addition to parameter types, etc., we must describe dynamic behavior (our 4th dimension).

Ben:

The models I build are normally quite reusable. It is up to the modeler, not the tool, to enable reusability.

Stefan:

I disagree with the statement. Or rather, whether you create a model that is easy to reuse or not is up to the modeler. If you foresee the need to reuse a model (or at least think there is a possibility for this), you ensure that you structure the model in such a way that this is possible in an appropriate way (and how this is done will vary from case to case). For starters, you always ensure that you use parameters for all quantifiable aspects in your model (and non-quantifiable too, if there is the slightest chance that this is something you want to experiment with—now or in the future). "Modularizing" the logic might also be a good idea every now and then. You "always" include more options to experiment with and more dimensions than the customer/manager "demands" because when modeling, you will often, after a while, know more about the challenges and system than the individuals with a "need" in the real world. And so on.
So to summarize, I claim this is really a talent that resides in the modeler, especially given a platform with such a high level of flexibility as AnyLogic.

Step-by-step into simulation modeling

Q 4. If I want to become an expert (like for PhD level) in AnyLogic simulation modeling, then what are the steps that I must go through? I already have basic Java and AnyLogic experience (Bayasgalan from Austria).

Andrei:

Participate in more real projects where simulation is used. Build real (=useful for somebody) models. It makes sense to look for an internship at an AnyLogic partner. Playing on your own with the tool is fun and will increase your language knowledge and debugging skills, but it won't help with the most important skill: conceptualization (=mapping the client's problem to the simulation model).

Otherwise, here's my list of skills for a good modeler:

- Communication, information, and learning skills.
- Knowledge of related and alternative technologies.
- "The skill/art of conceptualization."
- Minimalism and readiness to throw away your work.
- Striving for minimalism and readiness to throw away your work.
- Statistics and probability theory.
- Understanding of both continuous and discrete dynamics.
- Some coding skills.
- For agent-based modeling: some understanding of concurrency.
- Graphical design skills.

Don't be scared—you can do it, and it's a lot of fun! And if someone tells you something like "AI will do all that for you" or "No coding required," they are liars.

Ben:

Build, build, build. Only training makes you a master. So, find real problems to solve and do that for 1-3 years. You will then have seen the most common hurdles and can call yourself an advanced modeler.

Q 5. How to identify that you're stuck on the beginner level even when you're always learning something about simulation? When it seems you're upgrading, but in reality, you're stuck? (Matheus from Brazil)

Andrei:

I would only give you one hint here: if your creation (your model) appears to you as a "beautiful, elegant, and natural construct that had always been there, and you only happened to find it"—that's a good sign. If otherwise, your model looks like an ingenious, super-clever, complex piece of design—that's no good.

Ben:

Great question. You can gauge it somewhat by the leverage of OOP and Java. The more you apply that, the more you advance. Very experienced AnyLogic modelers use few of the blocks and much of the Java code. 😊

Q 6. If you were to draw the learning stages and curves of simulation modeling, what are the key concepts to elevate one's skills to the next? And do you aim to grow your technical foundation modeling and forefront visual communication simultaneously? Do we acquire one or the other through sheer practice? (Sebastian from the United States)

Andrei:

The visual part of simulation modeling always goes together with the logic part (in the extreme case of system dynamics, the logic equals the visual, but even there you can make the model look ugly or look elegant). Having good animation is not only about pleasing the management; it helps you with debugging the model, and it helps you (and the stakeholder) with understanding the system.

Stefan:

Answering questions 4 – 6.

Hard to answer! There are, of course, no clear phases, with clear-cut divisions between beginners, more experienced, and experts. It is a continuum, where you will benefit from experience. But experience is not everything, since you also need talent.

I have a few claims/thoughts related to what is important to evolve to be a good modeler:

1) Given the same experience, I claim that a modeler that has struggled with a more diverse set of challenges (preferably also in completely different industries) usually will become a more talented modeler. So, an individual that has worked both with transportation modeling, healthcare modeling, and production modeling will often be a "better modeler" than someone with the same amount of experience, having only worked with transportation modeling—also when it comes to a transportation challenge (of course, with exceptions). So, **diversity drives competence**!

2) Modeling is a creative activity—and an art/craft! It is also an activity where you have to be aware that the challenges you are modeling to address are dealing with uncertainty, not certainty. You create a model to make better decisions and navigate (hopefully better) towards a still unknown future. You therefore never should build your reasoning upon "data" (which usually represents the past, history), but always upon understanding the logic of the system you attempt to model. A **background in programming (preferably object-oriented) is much better than a background in statistics or data mining/handling**. And **experience from (and/or aptitude for) the applied and "real world" is much more important than an academic approach**. Modeling is an "applied science," not an academic one.

3) Related to the previous point, being a good modeler implies being **good at understanding challenges in the real world**. It is not the "technical modeling ability" that is most important, but the ability to encounter a real-world challenge, understand it, and see what possibilities there are to address/handle it. After you know that, then you know how a model might help in adding value.

So, modeling helps an individual to develop this competence (which also is why I am a bit skeptical related to letting AI play any major role here), but it is always good to have a bit of it from the start (or at least an aptitude for it). This ability to **understand the real world, the "business end" of the challenge, the relevant processes, KPIs, and parameters, etc.,** is

much more important than the more technical end, even though this, of course, in the end also is needed (to realize the wished-for model). A more senior modeler knows where to "put the boundaries" of the model, what to include or not, what to model in a more detailed or "rough" manner, what to visualize and not, and so on, better than a less talented one.

A summary of this bullet point is that the manager/customer "never" will understand modeling and simulation well enough to judge what should be done with simulation. But if the manager/customer only understands the business and real-world challenge and if the modeler only understands the technical end of modeling, then "the two will never meet." A model might be created, but it will seldom add value in the real world. Therefore, **the modeler (or someone leading the modeler, who also knows modeling) must have the competence to understand the real-world challenge so that this chasm can be overbridged**!

4) So, how do you know that you have evolved from being a less experienced modeler to a more experienced one? Well, when you start realizing that your customer (or manager) is really using the model or results generated by it. And when you start seeing that you add angles/dimensions to the modeling challenge that were not asked for (you **proactively** added this, based upon your own understanding), and your customer/manager afterwards sees the value of this. **When you are experienced, you seldom model what you are told or asked for—you model what you yourself know is needed!!!** But to reach that "level of cockiness," you need a certain level of understanding and experience/talent.

Modeling behaviors

Q 7. The question I have is how we model and simulate abstract concepts and behavior like policy, strategy, economics, social infrastructure, and natural environments? (Craig from the United Kingdom)

Andrei:

Read the book "Business Dynamics" by John Sterman. Although the System Dynamics method is used throughout the book, I suggest reading it regardless of the modeling method you are using. It is the most useful book on simulation modeling published to date.

It addresses all those questions about modeling abstract concepts. And then, should you decide to use not SD but Agent-Based modeling, read the theoretical chapters of our book https://www.anylogic.com/resources/books/big-book-of-simulation-modeling/.

Ben:

Like any other real process, really. There is no secret difference between a factory and climate policies. Only different assumptions and data.

Stefan:

It depends on the purpose of the model. For starters, you do not "model a policy or a strategy." What you might want to do is model a system (e.g., a healthcare system) that might be affected by a policy/strategy and then let the strategy/policy (or traits within the strategy/policy) be an alternative for a specific parameter. That way you might, if you can describe this well enough, simulate and better understand the consequences of policy/strategy A, B, C, etc.

And for economics, social infrastructure, natural environments, etc., it begins with asking yourself, **"What is the reason I am contemplating this model?". Given the answer to that question, together with an understanding of how much time/effort you can give it and knowledge of the stakeholders that might be interested in the model/results (because that will affect how you visualize and choose output KPIs), you can decide in what manner to model.** How this is done is up to you, the artist! As long as you fulfill the objectives you have with the model, you have succeeded in creating a dynamic painting! So, for the more abstract issues, the talent of the artist is more clearly tested than with the more clear-cut and down-to-earth ones, according to my view.

I can also add another reflection, a more general one, but perhaps extra relevant in more abstract challenges. The purpose of a simulation project is "never" to produce correct results (whatever that is). If you just focus on listing several conclusions and "results" in a report, you will often fail. **The important thing is not whether the results/conclusions/output are "correct" or not; it is that they influence the decision-maker/customer/manager and that she/he trusts the output.** If not, the correctness and simulation project is a waste of time! And this is, of course, why the visualization (not just

animation, but just as much visualization of logic, parameters, and output) is so important since this often is the way to create respect for and understanding of the output.

I more or less claim that one never should carry out a simulation project without showing the model in action at the same time as the output/conclusions are presented. The one in charge of the modeling endeavor must be in the room to take the discussion, explain, and in some form show the model. So, remember that an important output of a simulation project is the discussion as such when considering the various outputs and scenarios. So have a mindset to **demand—before starting the project—that you must be in the steering committee (or management team) meeting when the results are discussed**. You should lead this discussion! This is a general reflection applicable to most projects. But as I wrote, it is probably extra relevant in the more abstract cases.

Q 8. What are the economic theories behind agent-based modeling and simulation? What are the best books to read to model markets and behavior? (Mohammad from Turkey)

Andre:

See my answer to the previous question #7.

Ben:

Don't get bogged down with books and theories. Especially in ABM, do it. Model an economy, make assumptions, and play. Still check the literature, but it should not be a blocker.

Stefan:

I am not aware of any "economic theories" as such related to ABM. **But I think a question like this boils down to the fact that there is no firm definition of what ABM is, and different individuals will put different meanings into this concept.** Andrei would definitely be the best one to answer when it comes to judging ABM from the current most common way of "defining" ABM (and in a way, he probably is one of the few individuals in the world who can claim to "drive the definition").

For me, **ABM is really nothing more or less than to have a bottom-up approach when you describe or model a system (to compare with DEM and SD, which have top-down approaches)**, and to some extent, agent-based modeling is very much related to general object-oriented programming/modeling (it is the same way to think—to structure the whole with the help of objects, modules, or agents). You focus on the "small components" in your system (the product, the resource, the transportation vehicle, the patient, the machine, the factory, etc.) and build the logic related to this component inside the object/agent. Then you have other logic, defining how the agents interact through linking constructs, the environment, and so on. And when simulating, the interactions of the agents make the system "emerge," a system that most likely would have been impossible to describe with a top-down approach.

In SIMULA/DEMOS, queues of various kinds were often used to make various agents (in DEMOS called Entities...!) interact, and the logic inside each agent was defined in a process-centric manner (which can be compared with DEM). So SIMULA/DEMOS was not ABM or DEM—it was both (or discrete modeling). In AnyLogic, state charts are quite often used to define the logic in an agent, but given the libraries, other constructs could just as well be used. The important thing to realize is that a real-world challenge (at least the more complex ones) almost always demands mixing bottom-up and top-down approaches in the same model (a healthcare model needs a top-down approach for the whole process, the wished-for but most likely a bottom-up approach to capture the individual decision-making related to the patients, doctors, nurses, etc.; a traffic model might want an ABM approach to model pedestrians/passengers, but a DEM approach to model trains or buses). That is why multimethod or paradigm-free modeling is what we really need (an opportunity supplied by AnyLogic).

So, when it comes to what books to read, my advice is really that you should not focus on books related to ABM, but rather on books that give you an understanding of the field of competence. When it comes to markets, you have books about marketing (where, of course, Kotler is a classic guru, but there are many more worthwhile books/authors). When it comes to behavior, it would be books about organizational behavior. By learning the disciplines (marketing, org. behavior, etc.) and theories/models related to the disciplines and then learning how to model a system with ABM or paradigm-free modeling, your talent as an artist will enable you to create models of value in that field. So, my main point is that you **do not need to look for specific books combining ABM and marketing (e.g.)**. With

modeling—given a platform like AnyLogic—you can create models related to any discipline, given that you know the discipline well enough and model well enough.

Digital Twins

Q 9. Can I use AnyLogic to create a digital twin for the simulation and optimization of daily operational activities in a cooperative logistic network? (Golamn from the Netherlands)

Andrei:

Of course, our clients have already been doing that for quite a while. AnyLogic enables not only inside-four-walls but also larger-scale logistics modeling (due to GIS map integration and agent-based modeling capability). See the case studies at https://www.anylogic.com/supply-chains/.

Ben:

Yes. It is in the name "AnyLogic." You can model anything.

Stefan:

Again, it depends upon how we decide to define and look upon "Digital Twin" (DT) as a concept. For me, there are two alternatives:

1) "Digital twin" can be seen as a parallel field of competence/application to simulation, where the two fields partly overlap.

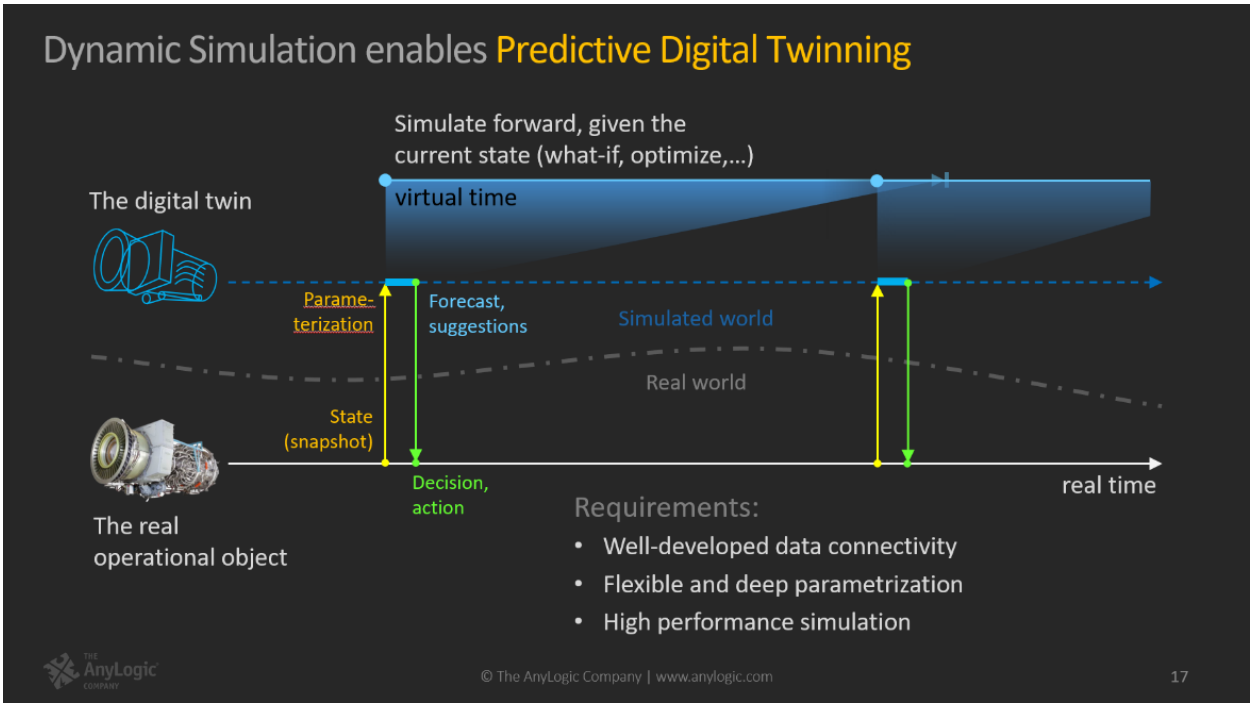
2) "Digital twin" can be seen as a subset of simulation, a specific category of model. For me, both points of view are acceptable. If we see DT as 2), it is obvious that AnyLogic can handle DT modeling. If we see DT as 1), it depends upon what we consider to be part of the overlap and not. But given the flexibility of AnyLogic (and the possibility to tailor-make the model with Java code), I would say that the "DT modeling challenge" most likely can be handled with AnyLogic.

So, this question clearly should get a yes as an answer, but I most likely wouldn't call that a DT. I would call it one of an almost infinite number of applications of simulation modeling.

Q 10. How does AnyLogic fulfill the definition of Digital Twin (not just a digital model or digital shadow)? There are many forms of Digital Twin models (e.g., physics-based, learning-based, etc.). (Sabah from the United Kingdom)

Andrei:

Here's our slide explaining Predictive Digital Twinning (as opposed to Informative Digital Twinning, i.e., shadowing the real system):



The AnyLogic software ecosystem for simulation (=AnyLogic IDE + AnyLogic Private Cloud) is fully compatible with Predictive Digital Twin requirements.

Ben:

Who cares? You can model any system at any time, past, present, or future, at any level of detail. Stop trying to fold stuff into weird definitions. Take your real problem and solve it with a model. 😊

Stefan:

I do not have a clear answer for this question, since I seldom/never use the term DT (I see DT basically as 2 above, and then I do not need the term), and the answer depends on how you define DT.

General questions

Q 11. Can we simulate a model by manipulating different variables to achieve its objective, then work backward to change the manipulated variables to make all of them realistic/close to reality while still achieving the same objective?
If yes, what is this approach called? (Ahmed from Italy)

Andrei:

That's called "Optimization experiment with constraints." A standard feature included in AnyLogic (and in other tools as well).

Ben:

Yes. Parameter variation.

Stefan:

In a way you do this, in some form, in many simulation projects. You have an objective, aiming at certain goals, and you iteratively experiment with describing different scenarios (by changing your parameter values), hopefully getting closer to the objective. I am not familiar with a term for this, however, and I might have misunderstood exactly what you meant.

On the other hand, it is a bit dangerous to think in terms of one objective. When we use simulation to address a challenge, we should almost always ensure that we have a portfolio of KPIs that covers various traits of the system modeled (a balanced scorecard approach). It is quite rare that all these KPIs turn green (get positive output) for a specific simulation/scenario. So, we seldom have "one objective," but rather a wish to balance several perspectives, where one perspective might be more prioritized, but the others must at least reach "an acceptable level." So, a simulation project can often be just as much about finding the weaknesses of a considered scenario (that still might be the chosen/"best" one) as it is about "maximizing a key KPI and reaching a specific objective."

Q 12. It seems that the key to advanced modeling is creating a level of abstraction that best fits the model's purpose (as simple as it can be, but no simpler). Then the ability to develop a good abstraction is closer to art than science. So how can you best develop that artistry? Is it a long modeling experience, an innate intuitive ability, domain knowledge, or a lucky discovery? (Mike from the United States)

Andrei:

Develop more real models (models that serve real purposes). Otherwise, this may be helpful:

- Always start with the simplest possible sketch.
- Always develop in iterations, including iterations with SMEs/stakeholders.
- Always follow Occam's principle, which in our context would be "if something (a variable, a parameter, an event, a shape, a block, etc.) is not absolutely necessary, it should not be there."

Here's a good picture:

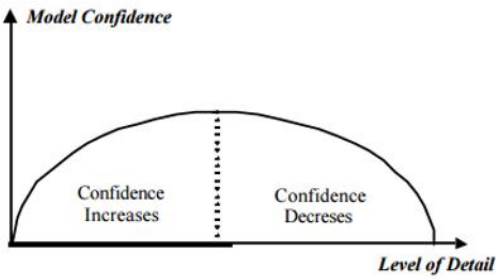


Figure 1: Relationship between "level of details" and "model confidence" (Lobão and Porto, 1997).

Ben:

Mostly failing, again and again. No luck or innate abilities are needed. Just patience.

Q 13. In AnyLogic, is a visual model "animation" a second step after making a model, OR does one build the animation in step with the model? (Logan from the United States)

Andrei:

Always build animation in parallel with logic.

Ben:

Always in parallel.

Stefan:

In AnyLogic you choose this yourself, and this is another "part of the art." So, in theory, you could start by modeling all your logic, and afterwards, you figure out a way to, in some form, animate. Or to not animate at all! The opposite would be hard (to first animate and then add logic), since you need to generate a course of events to have something to animate.
In practice, it probably often is best to handle this in a more iterative way. You start by adding some basic logic to your model, and after this, you animate in some form (probably quite roughly/simply for starters). After this, you expand the logic and probably also the animation. And so on. By having a modeling process like this, you also help yourself with quality securing your model, since the animation of a piece of logic can visualize logical errors. At the end, you might put more details into certain portions of your animation without doing anything further with the logic.

This is again an important aspect of the modeling platform you chose, since all alternatives do not allow this freedom to decide your own "modeling process." I think this possibility is crucial, and again, choosing the way you prefer to do it is part of the art. It should be possible to create a conceptual model without animation. It should be possible to animate in a very detailed manner, a cruder one, or not at all. It should be possible to animate early in the modeling process or not. AnyLogic allows you to think like this, to choose your own "modeling style." That is not always the case if you look at alternatives.