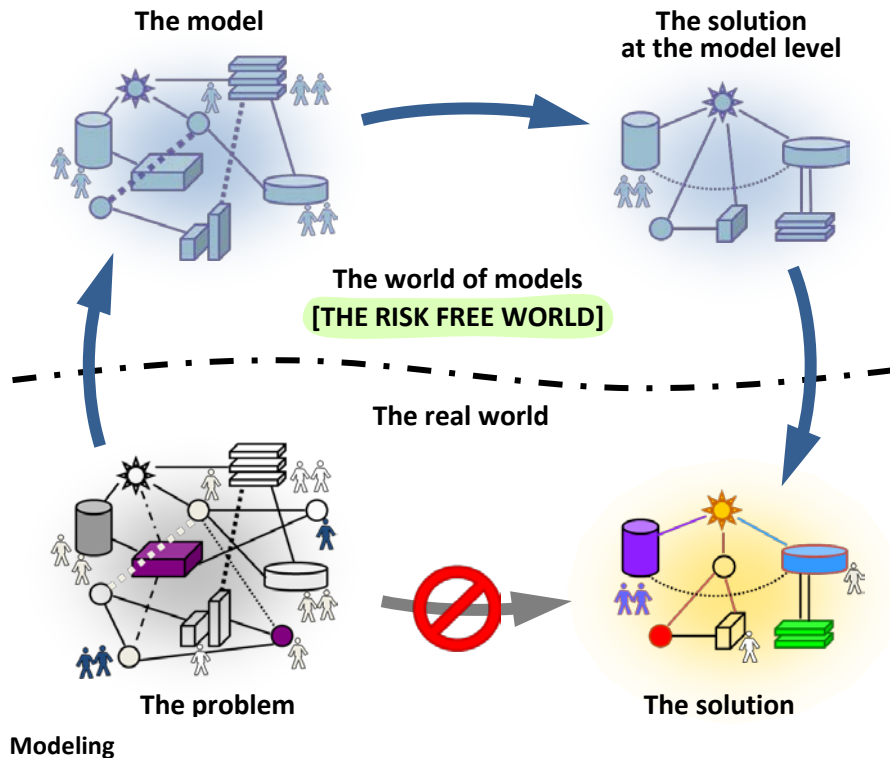


## Modeling and simulation modeling

In this chapter we will talk about modeling in general, types of models, and then focus on simulation modeling.

Modeling is one of the ways to solve problems that appear in the real world. In many cases we cannot afford finding the right solutions by experimenting with real objects: building, destroying, making changes may be too expensive, dangerous, or just impossible. If this is so, we leave the real world and go up to the world of models, see the Figure. We build a model of a real system: its representation in a modeling language. This process assumes abstraction: we throw away the details that (we think) are irrelevant to the problem we are trying to solve and keep what we think is important. The model is always less complex than the original system.



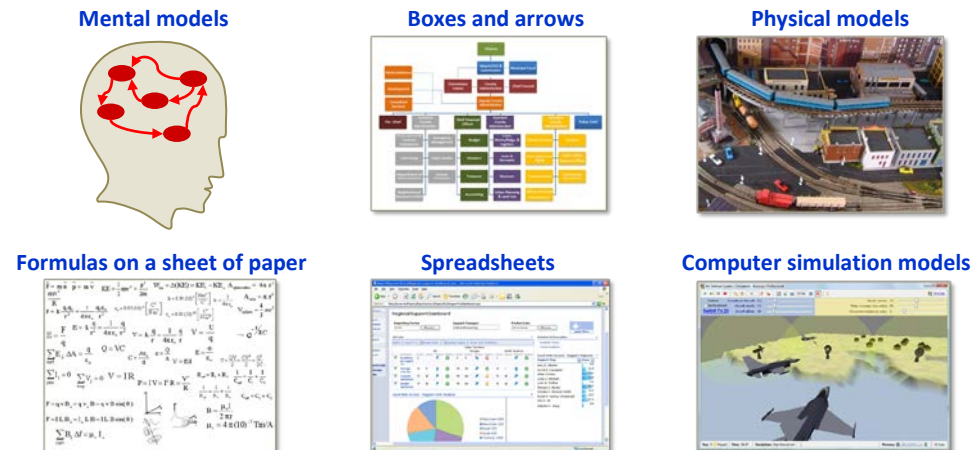
- ◆ The phase of building the model, that is mapping the real world to the world of models, choosing the abstraction level and the modeling language (= the method) is a less formalized thing in the whole process of using models to solve problems. This is still more an art than a science.

Having built the model (or sometimes even while building the model), we start to explore and understand the structure and behavior of the original system, test how the system will behave under various conditions, play and compare different scenarios, optimize. When we find the solution we are looking for, we map that solution back to the real world.

- 💡 **The whole modeling thing is actually about finding the way from the problem to its solution through a risk-free world where we are allowed to make mistakes, undo things, go back in time, and start all over again.**

## Types of models

There are many different types of models that we build. Consider Figure. Everybody builds mental models every day. A mental model is your understanding of how things work in the real world: friends, family, colleagues, car drivers, town where you live, things that you buy, economy, sports, politics, your own body. Decisions like what to say to your kid, what to eat for breakfast, who to vote for, or where to take your girlfriend tonight are all based on mental models.



### Types of models

An org chart of a company drawn using boxes with arrows is a model. You can use it to explain the structure of the organization, you can move people from one position or group to another and think about the advantages and drawbacks of a new structure.

If you take a pen and a sheet of paper and derive a formula for the optimal cross-slope of a road on a bend as a function of the turning radius and vehicle speed, you actually

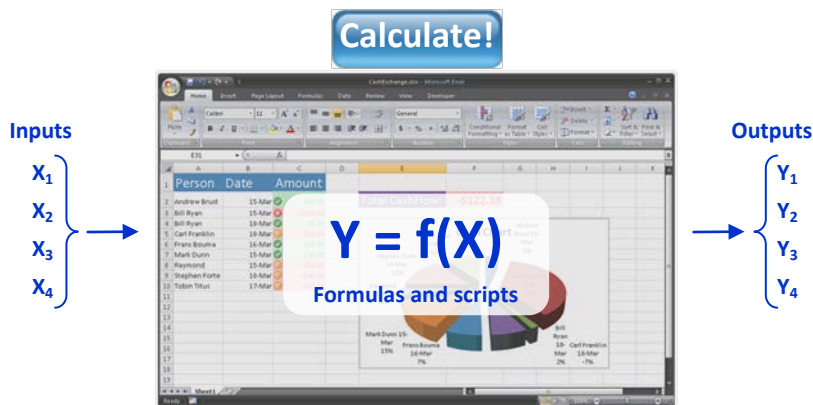
create an analytical model of vehicle movement in a turn based on another model – Newton's laws of motion.

Models can be physical. A model railroad can be used, for example, to optimize the layout and operation of a classification yard. A wind tunnel is a model of a free flight used to study aerodynamic forces and optimize the shape and design of the airplane.

Computers provide us with a flexible virtual world where we can easily create anything we can imagine. They are naturally and extensively used for modeling. Computer models can be of different kinds. The spreadsheet is the most accessible modeling software where someone can model arithmetic or algebra - such as expenses, for near foreseeable future. Software such as MS Visio™ can be used, for example, to plan your office layout; Autodesk 3ds Max™ to visualize interior design; Wolfram Mathematica™ to perform fast exact-match searches for sequences in the human genome; IBM WebSphere Business Modeler™ to model and analyze business processes. Finally, and this is the topic of this book, there are simulation modeling tools used to explore various dynamic systems from consumer markets to battlefields.

## Analytical vs. Simulation modeling

If you visit a group responsible for strategic planning, process optimization, sales forecast, logistics, marketing, project management, or HR management in a large company and see what kind of modeling tools and technologies they use you'll find out that the most popular modeling software is MS Excel™. Excel has obvious advantages: it is installed on any office computer and it is very easy to use. It is also extensible: you can add scripts to your formulas as the spreadsheet logic becomes more sophisticated.



Analytical model (Excel spreadsheet)

The technology behind the spreadsheet-based modeling is simple: there are cells where you enter the model inputs and there are other cells where you view the outputs. The output values are linked to the input ones via chains of formulas and, in more complex models, scripts. Various add-ons allow you to perform parameter variation, Monte Carlo, or optimization experiments.

There is, however, a large class of problems where the analytic (formula-based) solution does not exist or is very hard to find. This class, in particular, includes *dynamic systems* featuring:

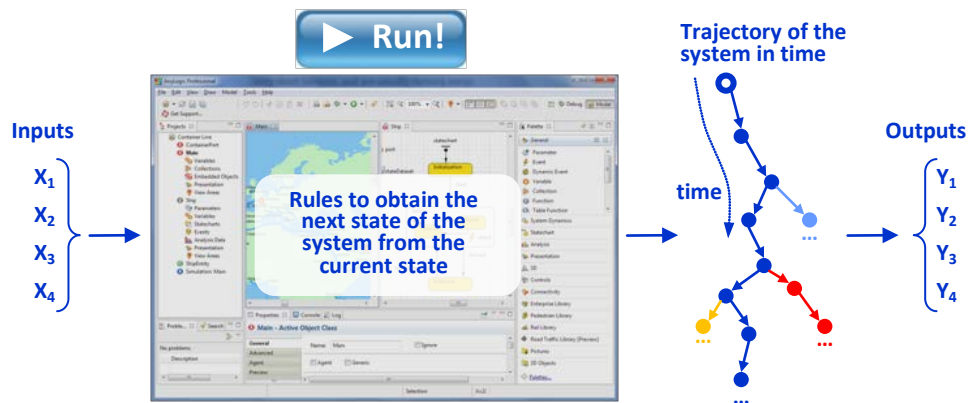
- Non-linear behavior
- "Memory"
- Non-intuitive influences between variables
- Time and causal dependencies
- All above combined with uncertainty and large number of parameters

You can't even put together a meaningful mental model of such a system not to mention assemble all the appropriate formulas.

Consider as an example a transportation optimization problem where you are to optimize the use of a rail car or truck fleet. Travel, loading and unloading times, maintenances, breakdowns, delivery time restrictions, terminal point capacities make that kind of problem very hard to approach with a spreadsheet. The availability of a vehicle at a particular location on a particular date and time depends on a sequence of events preceding that time. Answering the question of where to send the vehicle when it is idle requires the analysis of event sequences in the future.

- ◆ **Formulas that are good for expressing static dependencies between variables, fail to work when it comes to describing the systems with dynamic behavior. This is the time for another modeling technology that is specifically designed for analyzing dynamic systems, namely for *simulation modeling*.**

The *simulation model* is always an *executable model*: you can *run* it and it will build you a trajectory of the systems state changes over time. You can consider a simulation model as a set of rules that tell how to obtain the next state of the system from the current state. Those rules can be of many different forms: differential equations, statecharts, process flowcharts, schedules, etc. The outputs of the model are produced and observed as the model is running.



### Simulation model

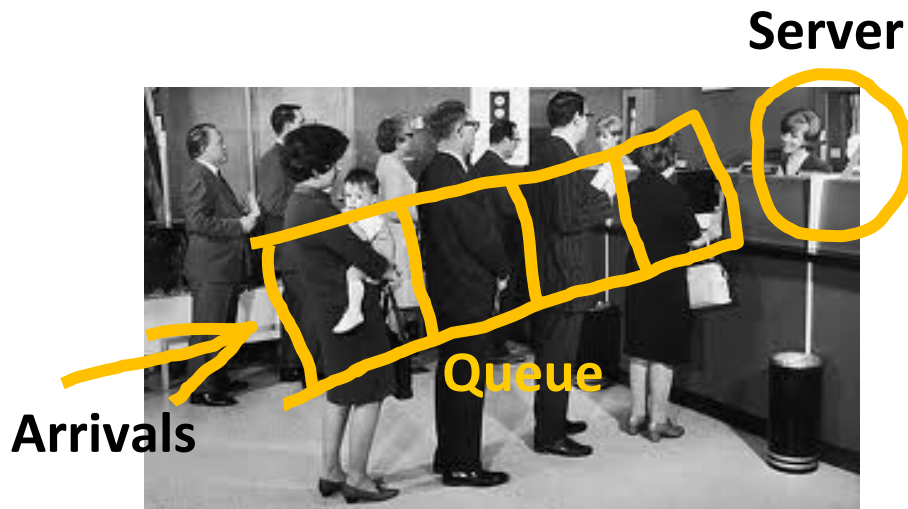
Simulation modeling is done with special software tools that employ simulation-specific languages, both graphic and textual. It typically requires some training and learning. But the efforts invested in adoption of simulation technology pay off when you need to perform high quality analysis of a system with dynamic behavior.

People (especially those who count themselves as Excel professionals and have some programming background), nevertheless, sometimes still try to build spreadsheet models of dynamic systems. As they feel the need to capture more details, they inevitably start reproducing the functionality of simulators in Excel. The models become huge and unmanageable. These monsters are full of code, they're slow, they have very short lifetimes, and they are usually soon discarded.

### The limits of analytical modeling: queuing theory

To illustrate the power of simulation and to better understand the limits of analytical modeling it is worth spending some time on **queuing theory**. *Queuing theory* is a mathematical approach to the analysis of dynamic systems with queues, such as computer transaction processing systems, call centers, transport, customer support, healthcare service systems. Queuing theory was mostly developed in the 1950s and 1960s before the computers became powerful enough to perform (resource-demanding) simulations. It addresses questions like: what is the average number of entities in the queue, what is the distribution of the waiting time, or what is the server utilization.

Consider an example: a bank. On average  $\lambda$  clients per hour enter the bank. At first we will assume there is only one cashier in the bank and on average he serves  $\mu$  clients per hour (mean service time is  $1/\mu$ ). We are interested in the client's waiting time, queue length and cashier utilization.



**A queue in a bank**

Queuing theory gives us an easy solution, see case  $M/M/1$  in the Figure. The formulas are very simple and give you the answer immediately. However, the formula for the waiting time is essentially based on two important assumptions:

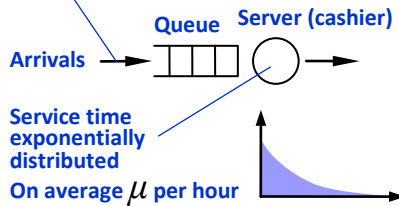
- A *Poisson stream* of clients, and
- Exponentially distributed service time

The first assumption means that the clients arrive at the bank independently, and the time the next client enters the bank door does not depend on the previous client. This looks like a fair assumption for the bank. However, the second assumption does not conform with reality. The distribution of the time spent by a customer at the counter should have some non-zero minimum, a major peak for the most frequent operations, and maybe a second peak for less typical operations (see case  $M/G/1$  in the Figure). The queuing theory does not give up and suggests another formula for the waiting time that is valid in case of arbitrary distributed service time: **Pollaczek-Khinchine formula**.

Suppose now that there is not one but three cashiers in the bank. This does not seem to be a big change in the service system. The analytic solution however starts look scary, see case  $M/M/K$ . And, moreover, it exists only in the case of exponentially distributed service time. For any other distribution there are no formulas.

And this is it. Any further complication of the bank service process does not have an analytic solution.

Poisson stream (independent arrivals)  
On average  $\lambda$  clients per hour



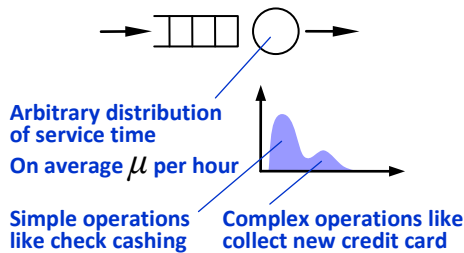
**M/M/1**

Server utilization\*:  $\rho = \frac{\lambda}{\mu}$

Average waiting time:  $W = \frac{\rho}{\mu - \lambda}$

Average queue length\*:  $L = \lambda W$  (Little's law)

\*These formulas are valid for all cases

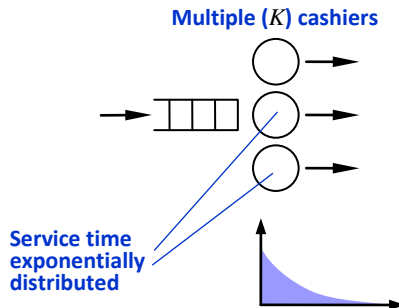


**M/G/1**

Pollaczek-Khinchine formula:

Average waiting time:  $W = \frac{\lambda(1 + C_z^2)}{2\mu^2(1 - \rho)}$

where  $C_z$  is coefficient of variation of service time

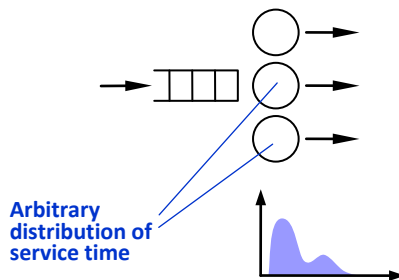


**M/M/K**

Average waiting time:  $W = \frac{P}{K\mu(1 - \rho)}$

$$P = \frac{(K\rho)^K}{K!(1 - \rho)} P_0$$

$$P_0 = \left[ \frac{(K\rho)^K}{K!(1 - \rho)} + \sum_{i=0}^{K-1} \frac{(K\rho)^i}{i!} \right]^{-1}$$



**M/G/K**

Analytic solution does not exist

Any further complication of the service process

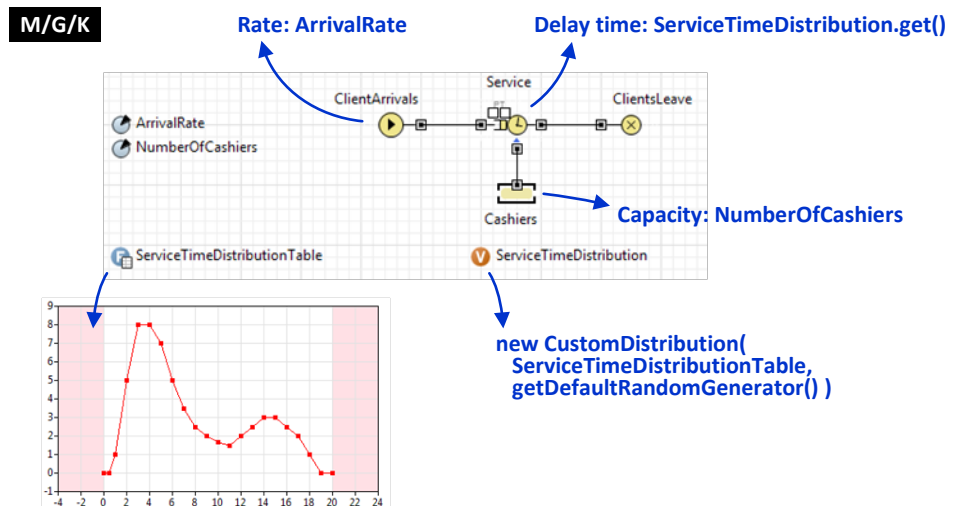
Analytic solution does not exist

Queuing models of a bank

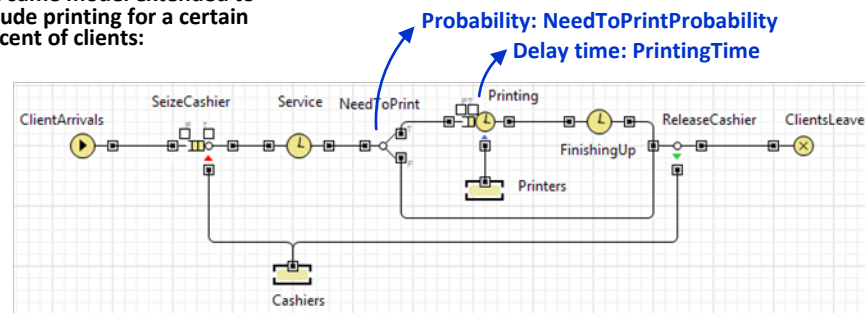
As you can realize, the process in a real bank is far more complex than even the M/G/K case, for example:

- Some transactions can be done only by some particular employees
- The client can be redirected from one employee to another
- The cashiers may share resources, such as a printer or a copier
- Different cashiers may have different skills and performance
- Etc., etc., etc., ...

Virtually any of those details are impossible to capture in an analytic solution. Even if formulas exist for a particular configuration, a small change in the process may make them void, and you will need a professional mathematician to fix them, most probably from scratch.



The same model extended to include printing for a certain percent of clients:



Simulation model of a bank



Simulation modeling, on the contrary, can handle service systems of any complexity. Simulation models scale well: adding more details to the service process or making a local change is captured by a corresponding incremental or local change in the simulation model rather than by re-creation of the model from scratch. At the top of the Figure you can see the simulation model for a bank with an arbitrary number of cashiers, Poisson arrivals and service time with the empirical distribution (M/G/K). At the bottom of the same figure the model is extended to include printing in a certain percent of cases and sharing a printer between cashiers.

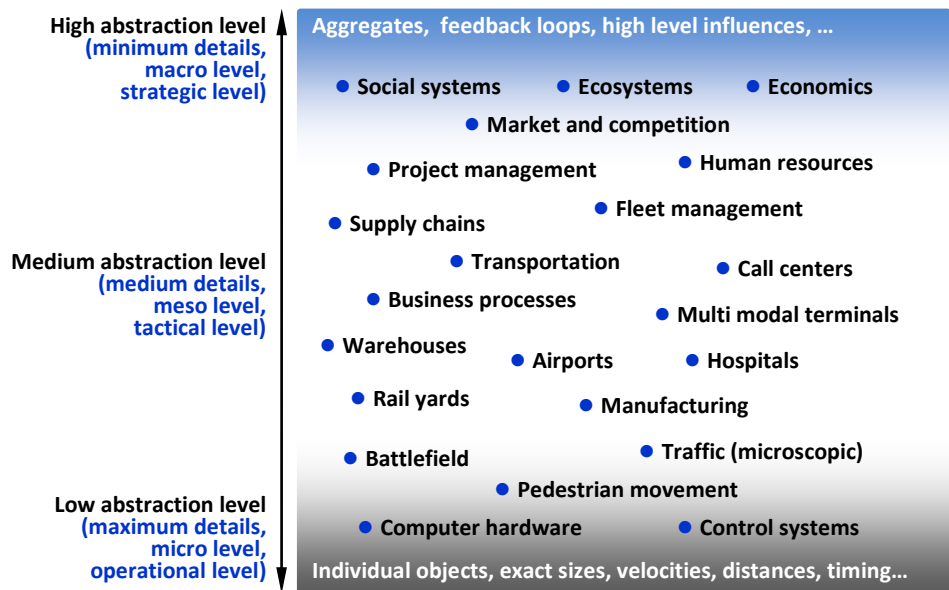
### Advantages of simulation modeling

There are six advantages to simulation modeling:

1. Simulation models enable you to analyze systems and find solutions where other methods (like analytic calculations, linear programming, etc.) fail.
2. Once you have selected the appropriate level of abstraction the development of a simulation model is a more straightforward process than analytical modeling. It typically requires less intellectual efforts, is scalable, incremental, and modular.
3. The structure of a simulation model naturally reflects the structure of the real system. As simulation models are developed using mostly visual languages, it is easy to communicate the model internals to other people.
4. In a simulation model you can measure any value and track any entity that is not below the level of abstraction. Measurements and statistical analysis can be added at any time.
5. Ability to play and animate the system behavior in time is one of the greatest advantages of simulation. Animation is used not only for demo purposes, but also for verification and debugging.
6. Simulation models are a lot more convincing than Excel spreadsheets (not to mention Power Point™ slides or reports with numbers). If you bring and run a simulation to support your proposal, you will have an advantage over those who bring just numbers.

### Applications of simulation modeling. Level of abstraction. Methods

Simulation modeling has accumulated a large number of success stories in a very wide and diverse range of applications. And, as new modeling methods and technologies are being developed, and as computer power grows, simulation penetrates new areas.



### Applications of simulation

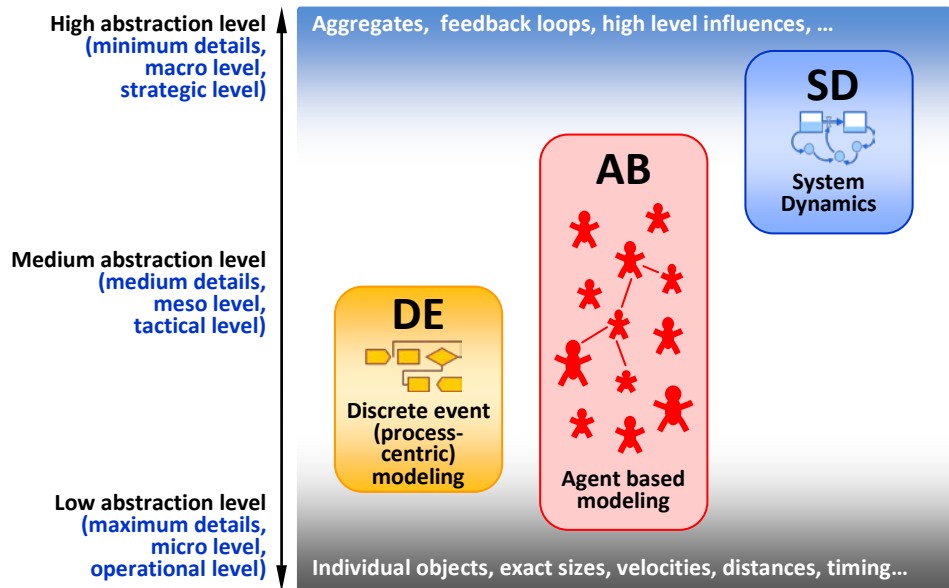
In the Figure some applications of simulation are shown sorted by the abstraction level of the corresponding models. The models at the bottom are physical-level models where real world objects are represented with maximum details. At this level we do care about physical interaction, dimensions, velocities, distances, timings. Anti-lock braking system of a car, evacuation of football fans from a stadium, car traffic at an intersection controlled by a traffic light, soldier interaction on a battlefield would be examples of problems that require modeling at a low abstraction level.

The models at the top of the chart are highly abstract. Individual objects are typically replaced there by aggregates. For example, instead of modeling each individual consumer we model the number of consumers, maybe divided into several categories; we model the number of jobs instead of individual jobs, etc. Correspondingly, interaction between the model objects is raised to a high level. In these models the amount of money invested into advertising may directly influence sales, and we do not model the intermediate steps in that causal dependency.

And there are models whose abstraction level is intermediate between low and high. For example, in a model of a hospital emergency department physical space may matter as we do care how long it takes to walk from the emergency care room to x-ray, but physical interaction between people walking in the building is irrelevant because we assume there are no congestions in the building. In a model of a business process or a call center we model the sequence and duration of operations and do not

care about space where those operations take place. In a transportation model we consider trucks or rail car's movement, loading and unloading, whereas in a higher level supply chain model we can assume that shipment of the order takes from 7 to 10 days and we do not care how the shipment is done.

- ◆ **Choosing the right abstraction level is critical to the success of the modeling project. Once you have decided what do you include in the model and what is left below the level of abstraction, the choice of the modeling method and the actual "coding" of the model is quite straightforward.**
- ◆ **In the model development process it is normal and even desirable to periodically reconsider the abstraction level. Typically you would start with high abstraction and add details as they are needed.**



### Methods in simulation modeling

In modern simulation modeling there are three methods, see the Figure. Each method serves a particular range of abstraction levels. [System dynamics](#) operates at high abstraction level and is mostly used for strategic modeling. [Discrete event](#) modeling with the underlying process-centric approach supports medium and medium-low abstraction. [Agent based](#) models can vary from very detailed where agents modeling physical objects to highly abstract where agent are competing companies or governments. The three methods are considered in detail in the [next chapter](#).