# 2 Agent-based modeling and simulation

This chapter presents basic terminology used in context of modeling and simulation in general, as well as in the area of agent-based modeling and simulation in particular. Within the remainder of this thesis the terms are used according to the definitions and explanations given here. After presenting basic terminology, this chapter discusses related work.

## 2.1 Basic terminology

The basic terms *(complex) system*, *modeling* and *simulation* are often used together and indeed, they are deeply connected. Nevertheless, it is important to distinguish these terms clearly in order to avoid misunderstandings.

### 2.1.1 Complex system

*Complex systems* are usually understood as systems consisting of a large number of heterogeneous, interacting components [93, 63]. The root causes of complexity are manifold and although more complexity metrics may be found, these are a few typical characteristics often attributed to complex systems:

- State space complexity (number of possible system states, number and value range of model parameters)
- Structural complexity (relationships and dependencies of system components)

R. Siegfried, *Modeling and Simulation of Complex Systems,*
DOI 10.1007/978-3-658-07529-3_2, 

- Behavioral and algorithmic complexity (intricate behavior and interaction patterns of system components)
- Temporal complexity (time- and state-dependent behavior of system components)

A key property of complex systems is that no single component controls the system behavior. Instead, the system behavior results from multiple and manifold interactions between the components. The term *emergence* refers to the fact that the system's overall behavior is not obviously derivable from the behavior of its constituting components. Interactions between the components have to be taken into account as well as effects of non-linearity [63].

### 2.1.2 Model

For this thesis a definition of a model is adopted which is not restricted to a specific domain:

**Definition 1 (Model)** A *model* is an idealized, simplifying and with respect to certain aspects similar representation of an item, system or some other part of the world. The purpose of the model is to allow a better study of specific properties than using the original system [54, p. 103]. □

In other words, a model is a goal-oriented description of a system that abstracts some parts of the original system with the intention to provide an easier explanation or analysis of the original system (cp. [16, p. 12]). Figure 1.1 illustrates this relation between the original system (i. e., the system under investigation) and the model as an idealized and simplified representation of that system. Depending on the purpose very different types of models are suitable to represent the original system. Once a model is developed, it may be explored or analyzed using different techniques, ranging from purely mathematical solutions to computer simulations. Depending on the chosen technique to solve a model, more specific terms are common, e. g., *simulation model* for a model which is solved by using simulation techniques.

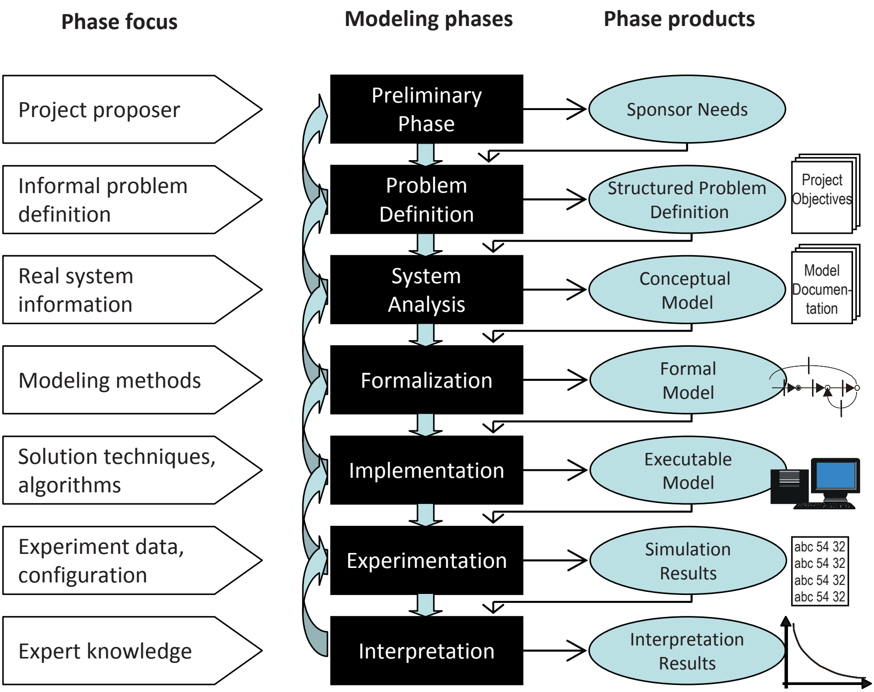


**Figure 2.1:** Generic model development process [77].

With regard to computer simulations, the results thus gained are generally of quantitative nature. However, the interpretation of simulation results with respect to the original system to answer the given questions may be quantitative or qualitative.

### 2.1.3 Model development process

In order to be precise, the term *model* has to be refined and augmented with additional information. In general, *the* model does not exist, but rather a model always exists in different stages. These different stages may best be explained following a generic model development process.

In [77, 106] a model development process is proposed that consists of seven phases (see Figure 2.1). The *Sponsor Needs* mark the beginning of each model development process. They document the original demand and requirements of the sponsor and describe the real problem in mind. Therefore, the Sponsor Needs are usually very specific with regards to the actual question in mind, but often miss information which are important for a model developer. Based on this initial description, the second phase focuses on creating a *Structured Problem Description*. The Structured Problem Description, which is created jointly by the sponsor and the model developers, takes up the information from the Sponsor Needs, augments them with additional information and organizes the information according to standardized templates.

The *Conceptual Model* is the main result of the system analysis. It contains all objects which are part of the model, defines their relationships as well as the properties and the behavior of all objects. Furthermore, the Conceptual Model defines the model structure (in terms of components and submodels). Formalization is the process of transforming a Conceptual Model into a *Formal Model* which is a detailed and formalized description of a model. At this point it is important to mention that if the Conceptual Model contains too few or too much information, the model developer has to iterate and extend or adapt the Conceptual Model first. This principle is also valid for the next phase: The *Executable Model* is a consistent and complete implementation of a Formal Model. Differences between a Formal Model and its corresponding Executable Model are nevertheless possible and sometimes unavoidable.

The Executable Model is finally used for experimentation and generates *Simulation Results*. These Simulation Results are usually unprocessed data (e. g., arrival times, object counters, etc.) gathered during simulation execution. In the final phase of the model development process the Simulation Results are interpreted by analysts and subject matter experts, thus creating the *Interpretation Results*. Ideally, these Interpretation Results answer the questions originally specified by the sponsor at the beginning of the whole process in the Sponsor Needs.

As already mentioned, there is no such thing as *the one* model. Instead, each simulation model exists in at least three stages (conceptual model, formal model, executable model). In colloquial speech the term *model* is often used synonymously with *executable model* and one has to be aware of the subtle differences.

### 2.1.4 Simulation

The term *simulation* is frequently used with two slightly different meanings. The first interpretation of simulation refers to the methodology of using simulation techniques for solving a specific problem. This covers the whole process of analyzing the problem, developing a simulation model (consisting of a conceptual model, formal model, and executable model), executing experiments and interpreting the results (see Figure 2.1).

The second interpretation refers to simulation as the act of actually executing an executable simulation model. Therefore, a simulation takes an executable model (and data) as input and applies a number of computational steps to transform a model from an initial state into a final state.

In the following, the term *simulation* refers to the second interpretation:

**Definition 2 (Simulation)** The term *simulation* refers to the execution of a specific executable simulation model. □

Although it is possible to simulate manually, almost always an appropriate simulation engine is used.

**Definition 3 (Simulation engine)** A *simulation engine* is a software application that executes the simulation of a model. □

A simulation engine may internally use any kind of data structures and execution control as long as the simulation is executed correctly. Within this thesis, the term *simulation engine* refers to a piece of software and not to the execution control mechanism or any other

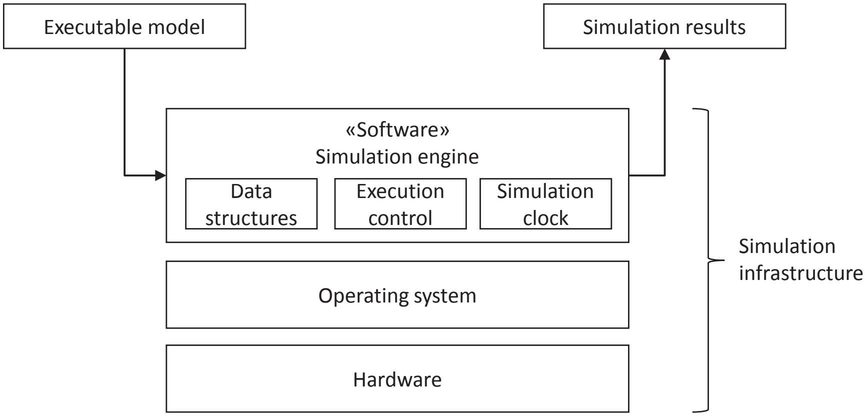


**Figure 2.2:** Components of a simulation infrastructure.

internal algorithm. The simulation engine together with necessary computing hardware and operating system is referred to as simulation infrastructure:

**Definition 4 (Simulation infrastructure)** The term *simulation infrastructure* refers to the entirety of hardware and software (operating system as well as simulation engine) necessary to execute a simulation model. □

Figure 2.2 ilustrates the components of a simulation infrastructure.

Using an analogy from a different area of computer science, the terms simulation and simulation engine may be explained as follows: Given an unsorted list, it may exactly be defined what is understood by *sorting* this list. However, sorting may be done by any sort algorithm, like bubblesort or quicksort. No matter which sort algorithm is used after its execution the list is sorted. Similarly, a simulation may be executed by different simulation engines. The important point is that the result is the same no matter which simulation engine is used, i. e., identical simulation results are produced.

Strictly separating the definition of the term *simulation* and the simulation engine as the actual software executing the simulation provides numerous benefits [151, p. 29]:

- Algorithms for executing a simulation as well as simulation engines may be specified and their correctness established rigorously.
- The same simulation model may be executed by different simulation engines, thus opening the way for portability and interoperability at a high level of abstraction.
- Different simulation engines may utilize the underlying computer hardware in an optimal way. Most notably, a simulation engine might parallelize the execution of a model.

Specific application areas of simulation may impose additional requirements on a simulation model or simulation engine (e. g., regarding update rate).

## 2.2 Agent-based modeling and simulation

### 2.2.1 Agent

Agent-based modeling and simulation is a paradigm which gains more and more attention for analyzing complex systems and becomes more and more widespread over the last years. While some authors claim that agent-based simulation seems to be a relatively new idea for the simulation community [109], others argue that agent-based modeling and simulation 'should not be seen as a completely new and original simulation paradigm' [27]. As the notion of an *agent* is the central idea upon which agent-based modeling is built, it is important to have a clear understanding of what is meant by this term. Surprisingly, there is no general agreement on a precise definition of the term *agent*.

Besides the ongoing debate and controversy, definitions tend to agree on more points than they disagree [148, p. 28], [83, 109, 80, 61]. For the purpose of this thesis, an agent is defined as follows:

**Definition 5 (Agent)** An *agent* is an entity that is *situated* in some *environment*, and that is capable of *autonomous action* in this environment in order to meet its *objectives*. □

This definition is very close to [148, p. 29] and conforms with the definitions given in [60, p. 280], [59, p. 83], [46], [61, 112] and [68]. In good accordance with this definition, the following characteristics of agents are generally agreed on [108, 83, 61, 112]:

- Agents are identifiable, discrete (and usually heterogeneous) individuals [93, p. 214].
- Agents are space-aware, i. e., they are situated in some kind of environment [83, 23, 120].
- Agents are capable of autonomous action and independent decisions. In this sense, agents are actively acting rather than purely passive objects [148, p. 28ff.], [93, p. 214], [68].
- In order to act within the environment and pursue their goals, agents are capable of perceiving their environment and acting within this environment [68], [148, p. 32].

Besides these generally agreed characteristics, many more definitions (partially very specific to certain domains) exist. Also, many more characteristics for describing an agent are available. Of these, at least one characteristic is worth mentioning. The *ability to learn* is part of many definitions of an agent and refers to the capability of an agent to adapt (and possibly improve) its behavior [93, p. 214]. The aspect of agents showing some kind of adaptive behavior or learning is not part of the definition of an agent used within this thesis. This is due to the fact that this requirement can not be applied to all kinds of agents (albeit to a huge number) and therefore is to restraining.

**Definition 6 (Agent-based model)** An *agent-based model* is a simulation model that employs the idea of multiple agents situated and acting in a common environment as central modeling paradigm. □

An agent-based model usually contains different types of agents which represent different individuals from the system under investigation. Multiple, distinguishable instances of each type of agent may be present in the model. This definition of an agent-based model does not answer the question whether agent-based modeling and simulation is something new or not, it rather stresses that 'agent-based modeling is a mindset more than a technology [22].'

Agent-based models are natural representations in social sciences [15] and thus many ideas stem from this area [22]. More generally, agent-based models are well-suited for systems with heterogeneous, autonomous and pro-active actors where individual variability cannot be neglected [120, 27, 83]. Furthermore, interaction between agents is usually regarded to be essential [27, 68].

Recalling the difference between a model and its simulation, it is now straight-forward to define the term of a multi-agent simulation:

**Definition 7 (Multi-agent simulation)** A *multi-agent simulation* is the simulation of an agent-based model. □

Similar terms frequently used in literature are: agent-based modeling and simulation (ABMS), multi-agent simulation (MAS), individual-based modeling (IBM), agent-based modeling (ABM), agent-based simulation (ABS). This thesis uses the term agent-based modeling and simulation (if only the model itself is referred to, the clause *and simulation* is omitted).

Although multi-agent *simulations* and multi-agent *systems* share many ideas (and are both abbreviated as MAS), it is important to distinguish precisely between these two terms. The main difference is that multi-agent simulations take place in a simulated world, whereas multi-agent systems are usually considered to have interactions with the real world.

### 2.2.2 Agent architecture

So far, only the characteristics of an agent itself have been described. Additionally, the internal structure and operation of an agent is important as it defines how an agent pursues and finally achieves its desired objectives.

**Definition 8 (Agent architecture)** An *agent architecture* specifies how the construction of an agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact (cp. [13, 85, 90]). □

By this definition, the agent architecture defines the internal structure of an agent, the component modules of an agent, their behavior and interactions [39, p. 447]. On an abstract level, the internal structure of an agent always consists of three main components [21, p. 10]:

- Sensor interface
  The sensor interface enables an agent to perceive the environment it is situated in.

- Effector interface
  The effector interface enables an agent to interact with the environment and to actively pursue its goals.

- Reasoner
  The reasoner is an internal component of the agent for processing the data perceived by the sensors, for decision making and for controlling effectors.

With respect to a specific agent architecture each of these components has to be detailed further, e. g., the reasoner of an agent architecture might include a knowledge base, workflow monitor, or planner [21, p. 10]. In summary, an agent architecture defines how sensor data (perceptions) and a possible internal state of an agent determine the next actions (effector outputs) and the future internal state of an agent [85]. This mapping of any given sequence of perceptions to an action is also referred to as *agent function* [112, p. 33].

For classifying agent architectures, various approaches have been suggested. According to Genesereth and Nilsson two general agent architectures may be distinguished [39]:

http://www.springer.com/978-3-658-07528-6

Modeling and Simulation of Complex Systems
A Framework for Efficient Agent-Based Modeling and Simulation
Siegfried, R.
2014, XIX, 227 p. 40 illus., 13 illus. in color., Softcover
ISBN: 978-3-658-07528-6