# IMPROVING SUPPLY CHAIN EFFICIENCY THROUGH SIMULATIONS - LITERATURE AND METHODOLOGICAL REVIEW

**Csaba Péterfi [1], Balázs Gyenge [2]**

[1]PhD student, [2]associate professor
[1,2]Hungarian University of Agriculture and Life Sciences
E-mail: peterfi.csaba.attila@gmail.com, gyenge.balazs@gtk.szie.hu

## Abstract

Increased competition, efficiency and profit orientation largely determine the operation of today's companies. While the XX. In the twentieth century, the emphasis was on mass production, large batch numbers, and economies of scale, until today, consumer demand has shifted significantly toward individualized products and services. The consumer expects almost perfect quality, reasonable price and immediately available stocks. That is why it is necessary to tailor the supply chain to the needs of consumers as perfectly as possible. Every small change in the way a company operates can have different effects that managers need to respond to immediately.

In a company where there is some flow of auxiliary fuels, semi-finished or finished products, the proper design of the supply chain is a focal point that largely determines its operation and efficiency, as logistics costs significantly affect the consumer prices of products and services. The solid foundation of my topic is provided by the logistics knowledge and the supply chain management that extends it, which already examines the entire supply chain instead of one player at a time. A XXI. The development of information and IT in the 21st century makes it possible to apply new approaches in this field as well and to develop an approach in which we do not assess the effects of changes in retrospect, but prepare for them in advance.

In my research work, the optimization of logistics activities through simulation is the focus. The simulation has been an integral part of technical sports and engineering work for many years, this methodology is not yet used in the field of supply chain management, which would provide an opportunity to analyze the effects of changes without risk and would be a tool for companies to model and measure their activities. And measurability would provide an opportunity for correction and improvement, which plays an essential role in increasing efficiency and effectiveness. Based on this line of reasoning, I believe that the combination of supply chain management and simulation holds untapped potentials.

In my article, I will review the currently known and useful simulation methodologies, which shows the processing of the issue and the prevalence of the methodologies.

***Keywords:*** *Supply chain, logistics, simulation, process development, efficiency improvement*
***JEL besorolás:*** *C63*
***LCC:*** *QA76.75-76.765*

## Preamble

Globalization and, as a result, there are no or very few products or services that are not available anywhere and from anywhere in the world. This ongoing compliance burden places a huge burden on companies. The rapidly evolving logistics and supply chain / network material flow and information flow processes are no longer just about the so-called basic processes (RST -

warehousing, transportation, storage), but the classic cost and time reduction goals seem to be constantly transforming along the supply chain. (Gyenge 2019)

Nowadays, logistics is no longer just a sales promotion activity, but in many cases it also determines the shopping experience. Just think about it, if we choose the ever-expanding way of consuming and shopping online, we want to take possession and use the product right away. The shopping experience is largely determined by the time the product is received. Companies are constantly trying to respond to needs, which can reduce lead times; they bring inventories closer to users, use fewer chain members in the supply chain, and so on.

In addition, in many cases we have the ability to follow the path of our shipment from the seller to the consumer, which is extra information and added value that was not important in the past and is essential today to maximize the customer experience.That is why it can be stated that nowadays it is no longer products and services that compete with each other, but entire supply chains, which can now be interpreted as a criterion for gaining competitive advantage, with the main goal of the supply chain maximizing value and meeting customer needs. (Kozma - Pónusz, 2016)

Logistics / supply chain management is an ever-changing, evolving field of science that has no choice but to take advantage of competitive advantages and meet the ever-increasing expectations of consumers by incorporating technical and technological innovations into its own operations and processes. In addition to more efficient operations, changes can have other effects throughout the supply chain that are unpredictable and can generate losses throughout the process. With the development of technology, we now have the opportunity to observe and study changes and developments with the help of simulation modeling. As Grygoryev says: Modeling is one way to solve real-world problems. In some cases, we can't afford to experiment with real objects and look for solutions like this: build, destroy, or if changes are too expensive, dangerous, or impossible. In these cases, we can build models, we can use modeling languages to represent reality. (Grygoryev, 2018)

**Formation and development of simulations**

The indisputable advantage of simulations is that we can observe processes after making changes without construction, destruction, and costly changes. Just as evolution is part of continuous adaptation and development to environmental change, economic actors face similar challenges due to changes in the economic environment. With the help of simulations, companies can prepare, what's more, benefit from change, avoid dangers, and take advantage of opportunities.

The development of simulations can be examined from several aspects:

- How to use
- Programming language and environment
- By areas of use,

however, David Goldsman Richard E. Nance James R. Wilson reviewed the development of the topic from an interesting approach. They took a series of events and discoveries that influenced the development of the simulations. (David Goldsman Richard E. Nance James R. Wilson, 2009)

 10.18531/Studia.Mundi.2021.08.04.27-40

The development of the simulation was divided into different stages:

As a first step, the pre-computer period was defined as:

***Precomputer era: From Buffon to World War II. (1777-1945)***

The Monte-Carlo method is usually derived from Buffon's needle experiment. The famous "needle-throwing" experiment, first proposed by Buffon in 1777, provides a good example of probabilistic modeling from a geometric perspective (Buffon, 1777). Suppose we have a large flat surface separated by a series of equidistant parallel lines. with distance d. (For example, we might think this is an alternating red and white striped field of the American flag.) An experimental ldd-length needle is thrown to the surface "randomly" and we want to calculate the probability that the needle intersects one of the parallel lines (e.g., the flag lies on both the red and white stripes). It is assumed that the surface is large enough for the needle to always touch it and that the boundary effects are negligible. Since Buffon's published solution contained an error which Laplace corrected in 1812, the Buffon-Laplace terminology is also used for the needle problem. (Laplace, P. S., 1812)

A century later, the simulations took on a surprising role in one of the most important applied statistical developments. William Sealy Gosset became a brewer at Arthur Guinness, Son & Co. Ltd. in 1899 at the age of 23. Guinness allowed Gosset to publish certain key statistical results, provided it used a pseudonym and no proprietary data was used. These results were published under the pseudonym "Student" beginning in 1908 with an article formulating the material now called Student Distribution (Student, 1908). Because Gosset's analysis results were incomplete, he used the crude form of the manual simulation to test his conjectures on the probability density function for the Student's t-distribution. This is the first application of the simulation in the industrial field active area.

The main problems in using the simulation include the strategic problem of designing a simulation experiment and the following tactical problems to run the simulations defined in the experimental design:

The start problem, i.e., determining when the simulation is in equilibrium (steady state), that any transients have died due to the initial state of the simulation;

• Estimation of the accuracy (variance) of steady state estimates based on simulation; and

• Perform accurate comparisons of alternative system simulations. In other words, what is the best set of alternative systems or system configurations in a competition?

For the boot problem, Conway (1963) proposed the first widely used rule for truncation (deletion) for observations contaminated with initialization bias generated by simulation.

Conway provided remarkable insight into the problems and solution strategies that would define the simulation for the next fifty years.

***Second phase: The formative period (1945-1970)***

Goldsman and colleagues believed that the rapid development of simulations in this era was due to two determinants:

• Construction of the first general purpose electronic computers such as ENIAC (Burks and Burks 1981);

• The work of Stanislaw Ulam, John von Neumann, and others using the Monte Carlo method on electronic computers in order to solve certain problems in neutron diffusion during the formation of the hydrogen bomb that were analytically unsolvable. (Cooper 1988)

In the 1950s, the proliferation and availability of electronic computers helped spread simulation techniques in various disciplines.

In 1960, Keith Douglas Tocher, a professor of operational research at the University of Southampton, developed the General Simulation Program (GSP), the first general-purpose simulator as a tool to simulate an industrial plant containing machines. (Tocher and Owen, 1960) Tocher's role in the '60s was outstanding and decisive, as he created the first textbook of simulations in 1963, The Art of Simulation (1963) and in 64, the activity-cycle diagram (ACD). ACD became the cornerstone of simulation education in the UK and the focus of research on program generators in the 1970s.

Long before his appearance in the American simulation language, Tocher devised and implemented an approach to combined simulation (discrete events and continuous model running) (Tocher and Splaine 1966). In 1960-61, Geoffrey Gordon introduced the General Purpose Simulation System (GPSS) as the Director of IBM Advanced Systems Development. GPSS is designed to facilitate rapid simulation modeling of complex remote processing systems, including, for example, urban traffic control, receiving and switching telephone calls. IBM's ease of use at the time and its software marketing policy made GPSS the most popular educational simulation language in the United States. A detailed description of the history of GPSS is summarized by Gordon. (Gordon, 1981, Káposzta, Nagy 2015)

In the 60s, the Simscript simulation program appeared. Based on FORTRAN, a common programming language (Backus, 1978), the initial 1963 version of the SIMSCRIPT simulation program was intended for users who were not computer experts and used a formula to define the model, initialize the model, and generate reports. The second generation, SIMSCRIPT 1.5, removed the limitations of FORTRAN, and SIMSCRIPT II served as a concept and idea generator, at the time it was considered the most ambitious programming language development. The published SIMSCRIPT II was a "layered" language with five defined levels. A comprehensive description of SIMSCRIPT is contained in the works of Markowitz (1979) and Nance. (1996).

Philip Kiviat, of Cornell University in the United States, has been instrumental in the development and implementation of SIMSCRIPT II. To Steel in 1961, where he developed the GASP (General Activity Simulation Program). Kiviat joined RAND Corporation in 1963 and became the main driving force behind SIMSCRIPT II. A successful version of GASP was developed with Alan Pritsker. RAND, IBM, Cornell, and U.S. Pat. Along with Steel, the Royal Norwegian Center for Computer Science was a hotbed for the development of the simulation language.

Kristen Nygaard and Ole-Johan Dahl started working with SIMULA in 1961. (Nygaard, 1961) With the strong support of Univac and a significant programming staff in addition to the two, SIMULA I was created as an extension of ALGOL 60, producing arguably the most influential programming language in computing. The history of the development of the simula is summarized by Nygaard and Dahl in 1981. (Nygaard and Dahl 1981)

The forerunner of the Winter Simulation Conference (WSC) in 1967 was the Conference on Simulation Applications Using the General Purpose Simulation System, which is now the most important international forum for disseminating the latest developments in system simulation. R. W. Conway, B. M. Johnson, and W. L. Maxwell of Cornell University laid down the central problems of digital simulation. Simulation problems were divided into two broad categories (Conway 1959):

• simulation structure and

• use of simulation

The problems in building a simulation model are as follows (Conway 1963):

• Modular design of simulation programs for easy reworking;

• Computer memory management;

• Checking for errors resulting from discretizing all continuous quantities inherent in digital simulation;

• Design and implement an efficient time management mechanism; and

• Manage files containing simulation entities.

Although many of the above problems have been resolved, the design and implementation of an effective time-saving procedure for dealing with certain types of events - even to this day - involves research and development.

***The expansion period (1971-1980):***

During this period, the specialists working in the field of simulation underwent continuous development of analytical tools in parallel with the evolving computing tools. With respect to discrete event computer simulation modeling languages, we immediately think of, for example, Pritkser and Hurst GASP IV; Kiviat, Villanueva and Markowitz SIMSCRIPT II.5; Pritsker and Pegden SLAM; Pegden SIMAN; Nance's conical methodology for object-oriented model development.

**Historical overview of simulation approaches**

After a historical review of the simulations - from which we learned what discoveries and mathematical methods led to the modeling of our present age - I would now like to bring together a variety of modeling forms and methods based on the work of Roberts and Pegden 2017.

Over the past 50 years, simulation has increasingly become a tool for analyzing operational processes. With the development of technology, simulations also developed in different ways, according to principles, and became tools for research and development and application. According to Roberts, simulation modeling is the part of simulation problem solving that serves to develop the model. This means the interpretation of a real problem in terms of a simulation language in which we are able to map, observe and solve reality.

However, the use of simulations requires some expertise, which I have formulated as follows (Kiviat, 1967):

• Correct problem definition and setting of objects.

• Use modeling concepts to abstract essential features of the system as a model.

• Collect and compile data and inputs for the model.

• Convert the model to computer-readable code capable of simulating the system.

• Instruct the computer to perform the simulation correctly and efficiently for different scenarios.

• Summary and analysis of simulation output in power meters.

Simulation languages have changed over the years, however, simulation approaches are similar. The simulation language executes a system model to dynamically influence the behavior of the system over time while changing the value of state variables over the simulated time.

Simulation languages can generally be divided into two parts:

1. Discreet

Where the state of the system changes in discrete units during specific events

2. Continuous

Where the state of the system changes continuously over a specified period of time. Today, most simulations are multimodal and support a modeling paradigm where discrete and continuous capabilities are mixed.

***Modeling World views***

During the 60-year history of the simulation, four different worldviews became used and dominated:

• Event

It provides the most flexibility, but is harder to use.

• Activity

Gives more context to events with some limitations.

• Process

Easier to use, but reduces modeling flexibility.

• Object

It is the simplest and most natural, but it limits the complexity of the model.

These views were developed by pioneers in the 1960s, (Gordon 1961; Markowitz et al. 1962; Tocher 1963; Dahl and Nygaard 1966) they have changed significantly today, not the basic ideas. They focused on achieving a balance between ease of use and flexibility when developing modeling tools. The development of simulation languages can be traced back to making process and object views more flexible while maintaining ease of use. However, modern simulation languages combine different methods and operate as a multimodal system.

***Event-driven modeling***

Events are consecutive points in time at which the system makes changes. It is the responsibility of the modeler to ensure that events are recognized and state changes occur. If the event occurs, the transition from the present to the offspring should be mapped. The data obtained should be collected and displayed.

Event-driven modeling focuses on a calendar of future events that shapes all future events. The operation of the discrete event simulation language can be described as follows:

1. Removes the next event from the calendar and updates the simulation time to that time

2. Performs the status update procedures for that event

This logic can be implemented in any widely used programming language, such as Fortran, C ++, C #, and so on. The functions of the programming language greatly facilitate the development of the simulation program. SIMSCRIPT II.5 added the basic options for defining entities, assigning attributes to entities, and collecting entities into sets. In the first 20 years of the simulations, the event-driven worldview was very common due to its flexibility and usability, however, the models were difficult to understand and required a high degree of proficiency from the user. A common method for visual modeling of discrete events is the event graph introduced by Schruben (Schruben 1983), which is a very useful and valuable tool, but its general use is severely limited.

***Modeling with activities***

Pidd (Pidd 2004) describes a model related to the three-phase approach called activity acquisition.

The essence of the approach is that activities form the basis of state changes when events occur. The three-phase method has evolved, but the activity scan consists of the following steps:

1. Advances the time to the next event

2. Processes one or more subsequent connected events and

3. It processes all other operations that depend on the occurrence of bound events.

While there is a need for a calendar of future events, there is also a need for a list of both tied and conditional events. These subsequent state-dependent (state-dependent) event lists are rescanned until no activity is started. Central to the three-phase approach is the use of a conceptual modeling tool that is now used as an “Activity Cycle Diagram” (originally referred to as a “wheel diagram”). Useful for identifying the activities of any system.

***Process modeling***

Process modeling appeared mostly in the form of GPSS (Gordon, 1961). Both a GPSS modeling tool and a simulation language developed at IBM. He saw the world as entities (transactions) that moved special purpose blocks in a complex model. Each block had a certain function. There is almost one-to-one correspondence between the block diagram and the simulation code. So the simulation model has a visual and written equivalent. This correspondence is important because the GPSS modeler can essentially translate the visual model almost directly into instructions, and instead of linguistic syntax, visual semantics has become the primary tool for modeling. This approach meant that the simulation modeler did not have to be a programmer, and so GPSS introduced a large number of people into the world of simulation who would not otherwise have participated in it.

It worked as follows:

In GPSS, the simulation performed two lists: a future event calendar and a current event calendar. When entities are created, they move through the block diagram as much as they can. If they encounter a block that may schedule their departure a little later, the entity is added to the future event calendar. However, if the entity cannot continue (resources may not be available), the entity is added to the current event calendar. The current event calendar is conserved and retrieved until no further action is taken (similar to scanning an activity), then the next event is removed from the event calendar and the time progresses.

An improved version was released in 1983 and several new process languages were developed. Further developments in the 1980s included better random number and random variant generation, variance reduction, efficient calendar management, and more efficient collection of statistics.

***Object-oriented modeling***

Another view of process modeling is that the model is a set of interacting processes or more generally objects. Developed in the 1960s, Simula (Dahl and Nygaard 1966) provided an early implementation of the idea of objects as elements of simulation, and that these objects could contain logical operations that control this object and that could interact with other objects in a synchronous or asynchronous manner. objects. Object-oriented simulation modeling usually falls into two camps. The first, exemplified by Simula, is that the simulation language should contain object-oriented concepts that allow the modeler to develop advanced simulation programs. Concepts relevant to this approach, such as abstract data types, inheritance, polymorphism, composition, parameterized types, etc., allow for a wide variety of objects and behaviors. The models are compact, efficient and expandable. In other words, they provide a better environment for simulation programming. Today, models are usually built in C ++ or Java in the context of simulation packages. The ideas introduced by Simula provide the basis for some recent developments in simulation for language designers to use an object-oriented approach to modeling simply easily and flexibly.

The ideas introduced by SIMULA are the most significant advancement in computing in the last 50 years, according to Roberts. The other object-oriented simulation modeling camp sees it as consisting of a wide range of predefined objects, each with a set of behaviors that are thought to be relevant to the control and use of the objects. This modeling orientation simplifies the model building process by offering a more natural and in many cases easier to use modeling paradigm. In the object-based approach to modeling, we create our model by placing software objects in our model that represent the physical components of the system.

In object orientation, the modeler simply describes the physical components of the system and the behavior and actions already built into the objects for those objects. Therefore, a working object has a predefined behavior that allows it to interact with the machines in the model and other workers. It's hard to imagine a more natural way to build a model than using a collection of pre-built modeling components that mimic the components of a real system. The challenge with this approach is that if we want to model anything in the real world, we need an extensive library of objects to be able to capture the vast variety of real objects you may encounter.

Due to its flexibility, process modeling has remained widespread, however, there has been an increasing emphasis on object-oriented simulation over the past 20 years. Newer object-oriented tools have rich object libraries that focus on specialized application areas such as manufacturing, supply chain, and healthcare. Simula also introduced the concepts of behavioral classes and subclass objects as part of an explicit modeling paradigm. The innovative work of simulation language design is realized through object-oriented simulation tools. Its advantage over process and event-oriented simulation is that we can assemble the logic of our model and the associated animations in one step.

***System dynamics model***

Systems Dynamics is a modelling approach developed at MIT in the late 1950s by Jay Forrester (Forrester 1961). It is a continuous form of simulation where variables vary continuously over time. System dynamics is sometimes used to approximate large-scale discrete systems (e.g., population modelling). In its general form, system dynamics has state variables that are dynamically related to a set of differential equations. However, for most applications, models consist of 'levels' and 'ratios', where levels are simply state variables and ratios are first-order differences (Sterman 2000).

System dynamics models are commonly used to represent stock and material flow diagrams. Causal loop diagrams are visual representations of the structure and behaviour of a system. The main simulation language for system dynamics was Dynamo, nowadays the common languages are Stella and Vensim. PowerSim and AnyLogic include system dynamics components, but also include event and process model features. We think of system dynamics models as continuous models, but the majority of applications involve modelling large discrete systems with many entities.

***Agent based modelling***

Agent-based modeling extends the concept of objects to agents whose properties are related to human behavior. In this way, the agents have intelligent independent properties and are able to make decisions. Although agents are independent, they are surrounded by other agents, so there are rules that govern individual decision-making and interactions between agents.

The functional structure of agent-based modeling is as follows (North and Macal 2007):

Agents are placed in the system and then allowed to evolve from the interaction of the agents. Each agent is a separate entity that interacts with other entities in the system. The focus is on modeling the behavior of the agent as opposed to the behavior of the system. In traditional process orientation, entities follow a series of process steps that are defined from top to bottom by the system. In contrast, agent-based modeling determines the local rules of behavior for each entity from the bottom up. Agent-based modeling is often implemented using an object-oriented modeling toolbar, so we do not view agent-based modeling as a new worldview, but rather an application that is implemented based on an object-oriented worldview.

A state diagram can be used to determine the framework of agent-based modeling and the behavior of agent classes (Shannon 1949). The state diagram can be used to determine the states that can be taken up by the agents, as well as the transient states that, when met, cause a transition between state pairs. Each diagram typically determines the behavior of an object in a particular class and the state transitions for object instances in that class. Although there are several versions of state diagrams, each defines states as nodes and defines arcs for the transition between states. The spread of agent-based modeling has become increasingly prominent since the 1990s, as with the development of computers, it is possible to manage the number of agents within a model, giving the user a high degree of flexibility (Káposzta et al. 2017).

The most common agent-based tools are Swarm, NetLogo, and AnyLogic.

**Branch of simulation modeling**

After an overview of history and approach, I created a branch diagram of simulation modeling, with which I would like to mold worldviews and orientations into a more transparent form.

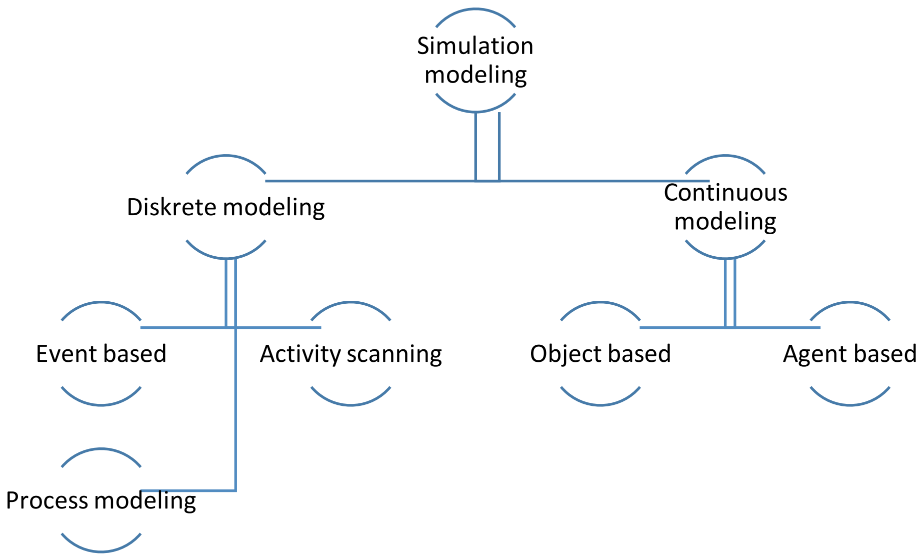


**Figure 1: modeling branch diagram**

*Source: Own editing (2021)*

**Simulation as a decision support tool**

The development, unbroken development, and brief history of the simulations testify to the immense need on the part of economic professionals to understand reality as accurately and accurately as possible. With the acceleration of technological development and the advent of industry 4.0, the importance of information as one of the most important raw materials becomes even stronger. Various methods are known for processing information, but simulations are extremely suitable for observing real-world problems. In a business environment, we may be able to make costly, dangerous, or impossible changes using simulations, the effects of which can be observed in cyberspace. Companies that choose to incorporate the use of simulations into their operations and thus be able to prepare for changes in the business environment can gain a competitive advantage over other market players.

Many companies have already begun to use simulation in operational processes as a decision support system, as Rozinat et al. (Rozinat et al., 2008) There is also a concrete example in the literature of how simulation can help increase efficiency. Salam and Khan (Salam and Khan, 2016) investigated a more efficient way of using the space of containers in maritime transport using simulation. Goldratt research labs (Goldratt research labs., 2018) conducted a study in which a steel plant was examined using simulation. Thanks to the comprehensive investigation and the improvements developed with the simulation, the processing capacity of the plant increased by 15%, which resulted in an increase in sales. The examples also show that the use of simulation can lead to efficiency gains and thus to building a competitive advantage, which is key for companies in today’s globalized world.

**Model building practice**

Professionals have recognized the value of being able to observe and experiment with how their environment works by modifying different variables. In the historical overview section, I listed the events that led to the development of the simulations. It can be said that it took several pre-age thinkers to build on each other in order for the simulations to develop. With the development of computers, various simulation modeling tools have also evolved. Their goal is to model reality in virtual space, which reflects as much as possible the processes you want to observe. In the theoretical overview, I reviewed the different principles, variants and their development of modeling tools. In this part of my research, I will present the steps required for model building through a practical example. Currently, we can talk about several widespread and used simulation tools that meet the expectations of korun. In my research, I also chose one of these tools, which is AnyLogic. In the AnyLogic simulation tool, we can create models according to different approaches, which can be:

- Agent based
- System dynamics and
- Discrete event-driven model

Thus, we may be able to examine the same problem from several aspects.

I will present the steps of model building according to the agent-based modeling principle. Agent-based modeling is a form of modeling that has been used for about 15 years, with the advantage of showing the area / process to be studied from a completely new perspective. Agent-based modeling does not have a general programming language, but rather the structure of the model consists of a graphical editing interface or scripts.

In practice, agents can take many forms:

- Person
- Company
- Vehicle
- Project
- Product
- Idea

Agents can communicate, influence each other's behavior, or, conversely, be isolated.

 10.18531/Studia.Mundi.2021.08.04.27-40

1. As a first step, we create a new model.

Then let's review the structure of the tree diagram as our model. By default, an agent class has a simulation experiment and a built-in database for reading the input data and writing the simulation outputs. In addition, we can use the runtime environment to upload the results to the cloud. Here we specify how customers get to the purchase.

2. Once we have reviewed the structure of our future model, the creation of agents can follow. Since we would like to create several similar objects, we select the agent population when creating the agents. We name the agents as buyers and then assign an appearance to them.

The parameters of the agents are then set. In our case, we focus on consuming an ad-sensitive product. We set the ad effectiveness parameter, which shows how many percent of consumers will be ready to make a purchase in a simulated day, to 0.1.We set the population size, which was determined at the beginning of the example to 5000 people. We then set the environment. In the Set New Environment menu, we create a default environment that scatter our agents based on an even distribution. After that, the model can be run, but until we set up the behavior of the agents, apparently nothing happens.

3. Defining consumer behavior.

The best way to determine the behavior of agents is with a state diagram.

Here you can define states and transitions. First, we specify the entry point of the state diagram, which is associated with the state of the user. We assign hearts to different states so that we can distinguish the potential consumer from the consumer. While running the model, it can be assumed that the agents become discolored during the simulation.

4. Visualize results using graphs.

The results can be viewed by including graphs. You can do this by specifying different functions, depending on what you want to show. We can fine-tune their results by also determining the following:

- Word of mouth effect
- Product life cycle
- Consumer impatience

Through this example, you can see what an agent-based model is suitable for in practice.

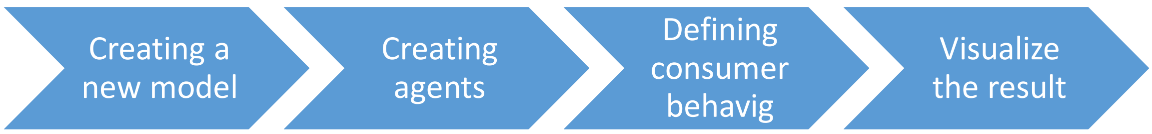


**Figure 2: Defining a model**

*Source: Own editing (2021)*

## Conclusions

In the present research, I have tried to summarize the historical events and research leading to the development of simulations, without which there would be no simulation tool today. After sequencing these, along with the development of simulation tools, I also reviewed different approaches. Finally, I presented the basic steps of model building through a market simulation example. From all this, it can be seen that simulation, as a decision support tool, can be very useful for a company to observe different situations in cyberspace.

## References

1. A. Rozinat1, M. T. Wynn, W. M. P. van der Aalst, A. H. M. ter Hofstede, and C. J. Fidge, 2008: Workflow Simulation for Operational Decision Support Using Design, Historic and State Information, file:///C:/Users/User/Downloads/Workflow_Simulation_for_Operational_Decision_Suppo.pdf
2. Backus, J. 1978. THE HISTORY OF FORTRAN I, II, AND III, IBM Research Laboratory, San Jose, California, ACM SlGPLAN Notices, VoI. 13, No. 8, August 1978
3. BUFFON, G 1777., Essai d'arithmetique morale, Supplement al'Histoire naturelle, 4
4. Conway, R. W., B. M. Johnson, and W. L. Maxwell. 1959. Some problems of digital systems simulation. Management Science 6 (1): 92–110.
5. Conway, R. W. 1963. Some tactical problems in digital simulation. Management Science 10 (1): 47–61.
6. Cooper, N. C., ed. 1988. From cardinals to chaos: Reflections on the life and legacy of Stanislaw Ulam. Cambridge: Cambridge University Press
7. David Goldsman Richard E. Nance James R. Wilson, 2009: A BRIEF HISTORY OF SIMULATION, Proceedings of the 2009 Winter Simulation Conference
8. Forrester, J. 1961. Industrial Dynamics. Vol. 1: Cambridge Mass: MIT Press
9. Goldratt research labs. 2018: Improving Reliability and Profitability of Integrated Steel Supply Chain with Simulation. https://www.anylogic.com/improving-reliability-and-profitability-of-integrated-steel-supply-chain-with-simulation/?utm_source=anl_mkto_email&utm_medium=email&utm_campaign=cs-manufacturing&mkt_tok=MDM4LU5GUC0zMzYAAAF80cKOcBy11jJ-bWln_mwVX3LvdebP2t_cF_bOfue3XnctLZaXVJqb6DRU5McBiKQ0a5tThrg303HWYKzLDPQFAZjUeRVw4lY0kdEcP-Nv5g
10. Gordon, G. 1961. "A General Purpose Systems Simulation Program". In Proceedings of the December 12-14, 1961 Eastern Joint Computer Conference: computers-Key to Total Systems Control, edited by W. H. Ware, 87-104. ACM.
11. Gordon, G. 1981. The development of the General Purpose Simulation System. In History of programming languages, ed. R. L. Wexelblatt, 403–437. New York: Academic Press.
12. Gyenge B., Mészáros K., Tari K., 2019; ÜZLETI INTELLIGENCIA (BI) ALKALMAZÁSA A LOGISZTIKÁBAN, Studia Mundi – Economica, Vol. 6. No. 2.(2019)
13. Ilya Grygoyev, 2018: Anylogic in three days – A quick course in simulation modelling
14. Kiviat, P. J. 1967. "Digital Computer Simulation: Modeling Concepts". RM-5378-PR, August, The Rand Corporation.
15. Káposzta, J.; Illés, B.; Nagy, H. (2017): Examination of impact of economic policy on quality of life in regions of some european countries with global perspective. ENGINEERING FOR RURAL DEVELOPMENT 16:1 pp. 236-241., 7 p. (2017)

16. Káposzta, J; Nagy, H. (2015): Status report about the progess of the Visegrad Countries in relation to Europe 2020 targets. EUROPEAN SPATIAL RESEARCH AND POLICY 22:1 pp. 81-99., 19 p. (2015)
17. Kozma T.- Pónusz M., (2016): Ellátásilánc-menedzsment elmélete és gyakorlata – alapok, Károly Róbert Kutató – Oktató Közhasznú Nonprofit Kft., Gödöllő
18. Kristen Nygaard. SIMULA: An Extension of ALGOL to the Description of Discrete-Event Networks. Proceedings of the IFIP congress 62, Munich, Aug 1962. North-Holland Publ., pages 520–522.
19. Laplace, P. S 1812: Théorie analytique des probabilités. Paris: Veuve Courcier,
20. Markowitz, H. M. 1979. SIMSCRIPT: Past, present, and some thoughts about the future. In Current Issues in Computer Simulation, ed. N. R. Adam and A. Dogramaci, 27–60. New York: Academic Press.
21. Mohammad Asif Salam Sami A Khan, (2016),"Simulation based decision support system for optimization", Industrial Management & Data Systems, Vol. 116 Iss 2 pp. 236 – 254, http://dx.doi.org/10.1108/IMDS-05-2015-0192
22. Nance, R. E. 1996. A history of discrete event simulation programming languages. In History of programming languages II, ed. T. J. Bergin and R. J. Gibson, 369–427. New York: ACM Press; Reading, MA: Addison-Wesley.
23. North, M. J., and C. M. Macal. 2007. Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation: Oxford University Press
24. Nygaard, K., and O.-J. Dahl. 1981. The development of the SIMULA languages. In History of programming languages, ed. R. L. Wexelblatt, 439–491. New York: Academic Press.
25. Pidd, M. 2004. Computer Simulation in Management Science: John Wiley and Sons Ltd
26. Roberts S. D., Pegden D. 2017: THE HISTORY OF SIMULATION MODELING, Proceedings of the 2017 Winter Simulation Conference
27. Shannon, C. E. 1949. "W. Weaver the Mathematical Theory of Communication". Urbana: University of Illinois Press 29.
28. Schruben, L. 1983. "Simulation Modeling with Event Graphs". Communications of the ACM 26 (11):957-963.
29. Sterman, J. 2000. Business Dynamics: Systems Thinking and Modeling for a Complex World. Boston, Irwin/McGraw-Hill
30. Student. (1908). The probable error of a mean. Biometrika, 6, 1–25.
31. Tocher, K. D., and D. G. Owen. 1960. The automatic programming of simulations. In Proceedings of the Second International Conference on Operational Research, ed. J. Banbury and J. Maitland, 50–68. London: The English Universities Press Ltd.
32. Tocher, K. D., and M. Splaine. 1966. Computer control of the Bessemer process. Journal of the Iron and Steel Institute 204 (February): 81–86.