sciendo

Transport and Telecommunication, 2025, volume 26, no. 2, 185-193
Transport and Telecommunication Institute, Lauvas 2, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2025-0015

EXTENDED ANALYSIS OF SIMULATION PARADIGMS AND DEVELOPMENT OF A MESOSCOPIC MODEL BASED ON THE DISCRETE TIME PARADIGM

Jurijs Tolujevs[1], Azat Zhumanov[2], Zhandos Kegenbekov[3]

[1]Transport and Telecommunication Institute
Riga, Latvia, Lauvas 2, LV 1019
[2]Al-Farabi Kazakh National University
Almaty, Kazakhstan, Al-Farabi Avenue 71, 050040
[3]Kazakh-German University
Almaty, Kazakhstan, Pushkin 111, 050010
[1]tolujevs.j@tsi.lv
[2]azat_jumanov@mail.ru
[3]kegenbekov@dku.kz

The theoretical part of the paper examines the paradigms of process simulation that have become standard by analysing the features of "time" and "state". The authors define two more paradigms of process modelling that have long been used in practice but are rarely mentioned in textbooks and scientific publications. As a result, the list of real-life paradigms of process modelling will look like this: Discrete-Event, Agent-Based, System Dynamics, Dynamic Systems, Discrete Time and Discrete Rate. The practical part of the work describes a simulation model of container train movement processes across the territory of the Republic of Kazakhstan. Although the model exhibits the features of the Discrete-Event and System Dynamics paradigms, the authors prove that it belongs to the Discrete Time paradigm. The selected level of detail for displaying processes occurring in a real system allows us to classify the developed model as a mesoscopic model.

Keywords: Simulation paradigms, discrete time simulation, mesoscopic approach, rail transportation, container trains

1. Introduction

For over 10 years, simulation specialists have been using the term "simulation paradigm," which is associated with AnyLogic Simulation Software (Borshchev, 2013). In most cases, three paradigms are mentioned in textbooks and scientific publications: Discrete-Event, Agent-Based, and System Dynamics. In principle, the AnyLogic package has always supported a fourth paradigm called Dynamic Systems. Each of these paradigms is dedicated to a separate Wikipedia article. The first three paradigms are well known to anyone familiar with the basics of simulation, but regarding the fourth paradigm, it is useful to know that it is referred to by the synonyms Continuous Simulation or simply Dynamic Simulation. The Continuous Simulation option most successfully emphasizes the features of this paradigm. The word "dynamic" can be applied to all paradigms in general, since they are focused on creating models of systems that change over time. It is known that in Continuous Simulation or System Dynamics models the time variable is changed using a fixed step "delta T". In Continuous Simulation models the value of this step is reduced as much as possible, since it affects the accuracy of the numerical integration result. In models of the System Dynamics type, this principle is usually not applied, since when choosing the "delta T" step, it is necessary to take into account a wide variety of features of the modelling task.

It can be argued that after the work of the German philosopher Hegel "Science of Logic" (German: "Wissenschaft der Logik"), which was published between 1812 and 1816, almost no one has written seriously about the concept of "discrete quantity" itself or about the inextricable connection between the concepts of "discrete quantity" and "continuous quantity". Of course, in mathematics there is a clear definition of a discrete variable, which can only take on certain values. Much less clear are the concepts of discrete and continuous time. Time can be called discrete only if a discrete variable is used to represent it in the model. In models where time is considered a continuous variable, processes are described by differential equations. Processes occurring in discrete time are described by difference equations. It follows

from these statements that when solving differential equations, continuous time can only be observed in analog computers, and in models on digital computers, only discrete time is used.

In publications, one can often find variants of interpretation of the concepts of discrete and continuous modelling that do not coincide with each other. Because the expression "discrete time moments" can be translated into other languages as "discrete moments of time" or as "moments of discrete time", some authors of publications and practitioners believe that Discrete-Event type modelling has such a name because events in the model occur in "discrete moments of time". It should be noted that this term does not make sense at all, since a moment in time cannot be continuous, and any fixed (recorded) moment in time can be considered discrete. The concept of "discrete event" is based not on the way time is displayed in the model, but on the way changes in the state variables of the model are displayed. The main thing here is the fact that at certain moments in time the values of the state variables change instantly, which is called an event in models of this type. In Discrete-Event models, events can most often occur at any point in time, and therefore such time should be called continuous, since no precisely defined values are specified for it in advance. Of course, discrete time can also be used in Discrete-Event models, when its values are calculated using a given step "delta T". In this case, it would be correct to say that events in the model occur at "moments of discrete time".

The theoretical part of the paper attempts to eliminate some of the above-mentioned uncertainties and contradictions by classifying models according to the "time/state" features. The authors define two more paradigms of process modelling that have long been used in practice but are rarely mentioned in textbooks and scientific publications. The practical part of the paper describes a complete simulation model that demonstrates easily interpretable results, but which cannot be unambiguously attributed to any of the three traditional modelling paradigms noted above.

2. Classification of models by "time/state" features

The type of model dynamics can be determined at a fundamental level by the behaviour of the variables t (model time) and S (state variable). Both variables can be either continuous or discrete. As a result, four main combinations are formed (Figure 1). In the case of continuous time, the t axis shows only random moments of time t1, t2, etc., when events occur in the model. In the case of discrete time, all time marks with a step of "delta T" are shown. The sign that the variable S is discrete is the marks on the coordinate axis S(t). If they are absent, the variable S is continuous, i.e. it can take any unlimited values. Let us consider in more detail the cases shown in Figure 1.

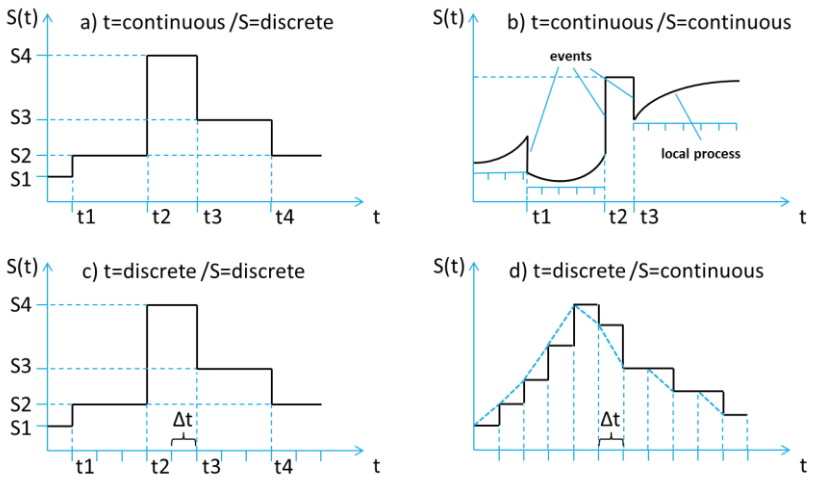


Figure 1. Four combinations of "time/state" features

a) t is continuous, S is discrete. This combination appears when using the Discrete-Event paradigm, when events can occur at any time, and the variable S is an integer, including the state number. A typical example is the models of queueing systems. For a service station, the model displays the states "free" and "busy". The number of occupied places in a queue or in some other storage is displayed as a non-negative integer.

b) t is continuous, S is continuous. This combination is also related to the Discrete-Event paradigm, but no restrictions are imposed on the variable S. For example, it could be the amount of bulk cargo in a warehouse, expressed in tons. Loading or unloading a car is an event in which the variable S changes abruptly. The diagram also shows local processes that use discrete time, and the processes themselves are modelled as continuous in state S. For example, the model can show the movement of cargo using conveyors. Such a local process is of type *d*, and the model as a whole in this case should be called hybrid.
c) t is discrete, S is discrete. Case a takes this form if events in the model can only occur at discrete time moments. In practice, such processes are rare. For example, we can assume that the gates leading to the warehouse territory are opened for a few minutes only at every hour. The variable S shows in this case the number of cars in the warehouse territory. The diagram shown in scheme *c*, also illustrates the "delta T" principle that is sometimes used to advance the time of a model based on the Discrete-Event paradigm. It is well known that in this case the time moments t1, t2, etc. may not coincide with the discrete time moments, so the events corresponding to them will be modelled with some delay.
d) t is discrete, S is continuous. All changes in such models occur at discrete time moments. This combination of features is the main one for models using the System Dynamics concept. The stepped line is suitable for representing the states of both flow and stock components. In the first case, the value S corresponds to the value of rate, and in the second case – to the value of level. Since flow retains the value of rate during the entire interval "delta T", the stepped line is ideal for displaying its dynamics. To show the dynamics of stock, it is better to use the dashed line shown in the diagram, in which the points on the boundaries of the intervals correspond to the value of level at the end of each interval. Such a line illustrates the process of continuous change of level in the corresponding stock model. The diagram shown in scheme *d* can be considered as a variant of the Discrete-Event process shown in scheme *b*, in which the "delta T" principle is used to advance the time of the model. This diagram also corresponds to the methods of solving differential equations using the Dynamic Systems paradigm.

As a result, it turns out that for the Discrete-Event and Agent-Based paradigms, schemes *a* and *b* are most often suitable, but sometimes schemes *c* and *d* are also suitable, and for the System Dynamics and Dynamic Systems paradigms, only scheme *d* is suitable. It should be borne in mind that each of the above process schemes in a specific model can be applied only to one specific state variable S. In relatively simple models, the dynamics of all state variables can correspond to one of these schemes, and in complex models there can be several such schemes.

3. Discrete Time modelling paradigm

In models using the Discrete Time paradigm, time changes with a constant step of "delta T", as shown in Figures 1c and 1d. By analogy with System Dynamics models, all changes in the model are tied to discrete time moments. The fundamental difference between this type of models and System Dynamics models is that their development is not based on the use of flow and stock components. The model developer can create any algorithm that describes what conditions should be checked and what calculations should be performed at each step of the modelled process. The use of the "delta T" principle in System Dynamics and Dynamic Systems models is associated with the need to perform numerical integration of differential equations. When working with the Discrete Time paradigm, such a task does not arise. In some publications, the concept of Discrete Time is associated only with the above-mentioned time advance method in Discrete-Event models, but increasingly it is interpreted as a separate modelling paradigm.

The book by the "classics" of discrete modelling (Zeigler and Kofman, 2018) contains a separate chapter "Discrete Time Models and Their Simulators", which explains the method of constructing a modelling algorithm described above. The process of the model's functioning is briefly described as follows: at a certain point in time, the model is in a certain state, and it determines how this state will change, that is, what the state will be at the next point in time. It is also noted that the Discrete Time principle is used not only in the simulation of real systems, but also in the simulation of such mathematical models as Cellular Automata and Markov Chains.

The book (Tang *et al*., 2020) is devoted to the modern problem of counteracting computer hackers, but it contains a chapter on "Discrete Time Simulation". Along with the well-known abbreviation DES (Discrete-Event Simulation), the authors introduce the abbreviation DTS (Discrete Time Simulation). It is noted that DTS is more suitable for systems in which states change at uniformly distributed time points. The book discusses in detail the car-following model, which is based on simulating microscopic movements occurring under Discrete Time conditions.

The paper (Morrison and Diesmann, 2007) addresses the challenges of designing a Discrete Time modelling tool for neural networks. The complexity of the task is increased by the fact that the tool is designed to be distributed across many computers. The authors report the creation of a tool that supports the exchange of point events in networks of neurons in both discrete and continuous time.

In the paper (Buss and Al Rowaei, 2010) the authors discuss in detail the problem of the influence of the step size "delta T" on the accuracy of the modelling results. The authors come to a quite understandable conclusion that with a large step size "delta T" the model is executed faster than the reference Discrete-Event model, but the accuracy of the modelling results will be low. With a small step size "delta T" it is possible to obtain accurate results that coincide even with the results of analytical calculations, but the model processing will require more time.

In the MATLAB programming language, the concept of Discrete-Time Simulation is applied to methods of numerical integration of differential equations that belong to the Dynamic Systems paradigm, if we use the previously mentioned classification proposed in the AnyLogic package. A separate chapter on this topic can be found in each description of the MATLAB language, for example in (Toney and Jayakumar, 2022). An example of using MATLAB to develop conventional simulation models of logistics systems can be found in the master thesis (Riezebos, 2016). In this work, the apparatus of differential equations is not used, but the logic of specific processes that are implemented under Discrete Time conditions is directly programmed.

The ability to perform arbitrary calculations at each discrete time step makes it desirable to use spreadsheets for such modelling. The main advantages of spreadsheet simulation were described already in (Seila, 2003), where it was noted that the values of the model variables are conveniently displayed in rows, each of which corresponds to the next moment of Discrete Time. Calculations in the model based on MS Excel can be implemented both using macros written in the VBA programming language and simply Excel formulas. The first case is represented by the book (Elizandro and Taha, 2007). An example of the second case is the work (Love, 2017), which describes a model of a technological process, in the processing of which a time step of 10 sec is used.

4. Discrete Rate modelling paradigm and building mesoscopic models

In models using the Discrete Rate paradigm, events occur at any time, but between these events, all flows of some substance remain constant. If in such a model the state S is the content of a storage device, then the diagram in Figure 1b is suitable for its dynamics. Flows in such a model are characterized by the value of rate λ(t), which is measured as the amount of substance per unit of time. In this sense, Discrete Rate models are like System Dynamics models, but the rate for each flow is not related to the step "delta T" and can retain its value for an arbitrarily long time. Figure 2 shows the input and output flows, as well as the content S of a storage device that depends on them.

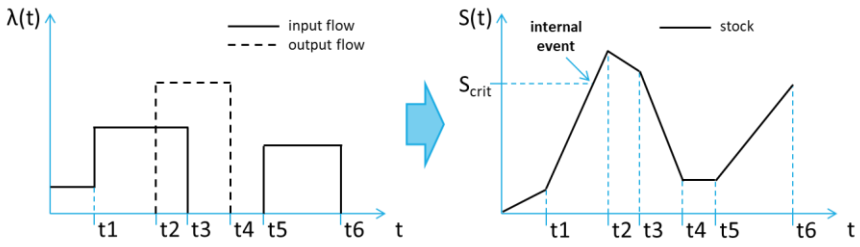


Figure 2. Dynamics of flows and storage contents in a Discrete Rate Model

Six moments of time correspond to external events when the input or output flow changes its rate. An internal event is also shown when the storage level reaches a given critical level S_{crit}. In Discrete Rate models, the event types "a given level is reached when moving up" and "a given level is reached when moving down" are defined. Any actions can be associated with each such event, for example, stopping or starting a flow. It should be noted that with a piecewise constant flow intensity λ(t), the S(t) diagrams will have only a piecewise linear character, as shown in Figure 2. This property of the S(t) diagrams is used to accurately calculate the time of occurrence of the above-mentioned event types.

The concept of Discrete Rate was introduced into the practice of simulation modelling by the developers of the ExtendSim package (Phelps *et al*., 2002). A very detailed description of the capabilities of this simulation method is given in the article (Damiron and Krahl, 2014), the authors of which are also

the developers of the ExtendSim package. The Discrete Rate paradigm was first implemented in software in 2007 in the ExtendSim version 7 package. Around 2015, the Fluid Library appeared in the AnyLogic package, which also allows for the full implementation of the Discrete Rate paradigm. The developers of both products focused primarily on modelling the processes of movement and storage of liquids, bulk cargo or large quantities of individual objects, such as bottles on a bottling line. However, by using the concept of mesoscopic modelling, it was possible to demonstrate the great advantages of this paradigm in the development of simulation models of a wide variety of material flow systems.

The concept of mesoscopic modelling has long been used in the field of traffic simulation, where it is defined as building models that are between microscopic and macroscopic models in terms of detail (Sun *et al.*, 2020). The work (Reggelin and Tolujew, 2011) describes the principles of building mesoscopic models of processes in logistics systems. Such models are more compact compared to microscopic models, require relatively small amounts of initial data and produce results that are accurate enough to assess the main properties of the simulated systems. The results of further development of these ideas are shown in (Hennies *et al*., 2014). In both publications, the Discrete Rate paradigm is considered one of the most promising approaches for building models of this type. An example of the application of this paradigm to traffic modelling can be seen in (Savrasovs, 2012), and for the production process – in (Reggelin *et al*., 2020).

5. Mesoscopic model of container train movement and handling in the Middle Trade and Transport Corridor

The model is built using System Dynamics principles, but it has the mapping of individual objects (container trains and railway ferries) typical of Discrete-Event models. The unit of time measurement in the model is one day. This means that for each flow of the model the value “the number of objects that passed through a given point of the model for one day” is defined, and for each storage of the model the value “the number of objects that are in this storage at the end of the next day” is defined. The storages in the developed model are waiting areas for trains or ferries, as well as all sections of the railway track that can accommodate a certain number of trains. Flows show the movement of trains or ferries between the related storages.

The model is implemented using the Vensim PLE program, which is oriented towards the System Dynamics method. The difference between the developed model and micromodels based on discrete events or agents is obvious, but the model has features that distinguish it from traditional macromodels created based on System Dynamics:

- the model's time scale is relatively short and shows only 30 days of the process (this number can be increased or decreased if necessary);
- what is being modelled is not a mass flow of a continuous substance, but the movement of several dozen discrete objects (container trains and railway ferries);
- a special method of model initialization is used, which allows to describe the continuation of processes started before the start of modelling.

Although the software implementation of the model is based on a tool that is traditionally considered a means of creating continuous models, given the set of properties of the created model, it can be considered that it corresponds to the Discrete Time paradigm. Since all model variables can only take integer values, their dynamics correspond to the diagram in Figure 1c.

5.1. Description of the conceptual model

The object of the modelling is the process of movement and processing of container trains in the Middle Trade and Transport Corridor (World Bank, 2023). Such trains arrive from China to the border stations of Dostyk and Altynkol (Khorgos station in Figure 3), pass along three routes through the territory of the Republic of Kazakhstan and depart from the ports of Aktau and Kuryk in the direction of the Azerbaijani city of Baku using railway ferries. Figure 3 shows the part of the Middle Corridor that is reflected in the developed model. The three modelled routes can be described as follows using the notations in Figure 3:

- Route 1: Dostyk – Zharyk – Saksaul – Port Aktau/Kuryk;
- Route 2: Dostyk – Zhetigen (Almaty) – Arys – Saksaul – Port Aktau/Kuryk;
- Route 3: Altynkol (Khorgos) – Zhetigen (Almaty) – Arys – Saksaul – Port Aktau/Kuryk.

The aim of the simulation is to obtain data on the locations of all trains and ferries on each day of the simulated process. The input data of the model describe the incoming flows of trains and ferries and set the parameters of the train handling processes at border stations and ports. These data are not generated

using random numbers but are set by the model user as precisely planned or expected. In the second case, the user conducts several experiments with the most probable values of the model input data. Thus, before each run of the model, the user creates a monthly scenario combining all the model input data. The main results of the simulation are presented in the form of diagrams, which show all trains in motion, processing or waiting for each day of the simulated month. These diagrams are used to analyse train movements and delays and to conduct experiments to assess the impact of infrastructure elements on these processes.

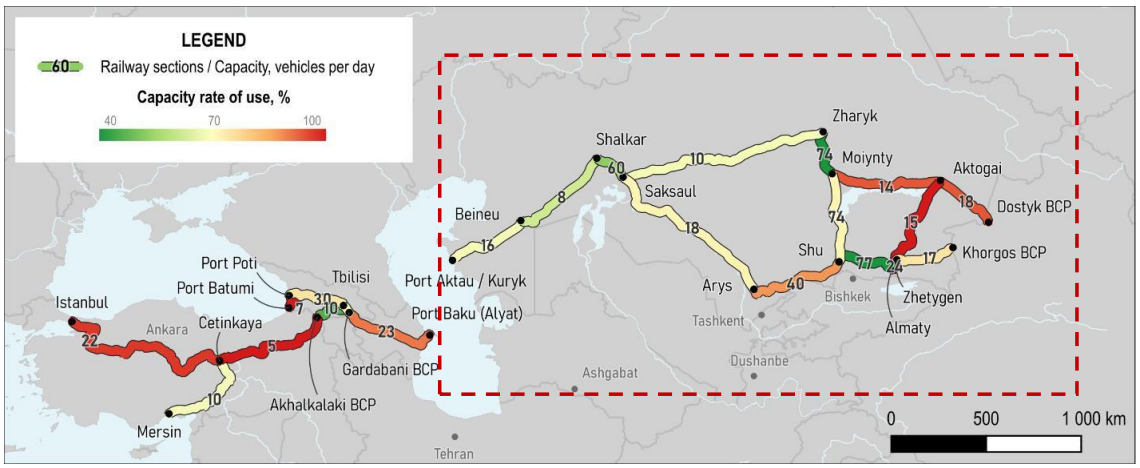


Figure 3. Selecting the modelling area on the Middle Corridor map (World Bank, 2023)

An important feature of the developed model is its initialization method, oriented to the Digital Twin technology. When the model is launched, it is not "empty", as it displays all the trains and ferries that are in the simulated system at the beginning of the day that will be the first in the simulation of the next 30 days. Moreover, the data for initializing the model are specified in a dynamic form, i.e. for all trains on the way, the expected times are indicated when they will appear at certain points on the route. This technology is focused on using data from transport corridor information systems in real time.

The choice of the mesoscopic level for displaying processes in a transport corridor is based on both the actual level of accuracy of the available initial data and the requirements for the accuracy of the modelling results. For example, only the number of trains expected on each day may be known about the flow of trains arriving at the first station of a route. The approximate time of train arrival in hours and minutes may be known only for the first day of the process, but in practice even this condition is often not met. The user of the model is not interested in the hours and minutes when some events will occur at the points of the route, since no model can calculate them with such accuracy. He is interested, for example, in how many container trains will be loaded onto ferries on the tenth day of the modelled process, given the actual schedule of ferry arrivals at the port of departure.

5.2. Implementation of the model using the Vensim PLE program

The model is developed based on the Vensim PLE simulation program (Vensim, 2024), and it has a modular structure, shown in Figure 4. In total, four types of modules are used in the model. A brief description of the Station Altynkol module is given below as an example. In all modules of the model, the constants or variables that are the input data or initialization data are marked in colour. Dynamic data that is entered into the model as a sequence of 30 numbers are marked in red.

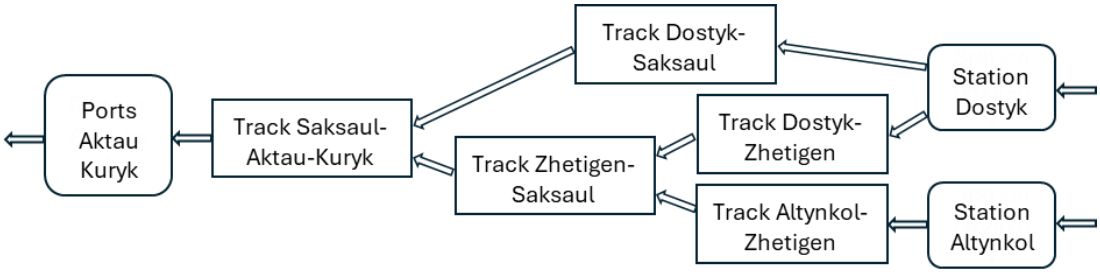


Figure 4. Modular structure of the simulation model

Figure 5 shows the structure of the Station Altynkol module. The input stream "container train input Altynkol plan" refers to the dynamic initial data and is specified by the user as the expected number of trains in each of the 30 days. The variable "percent delay Altynkol" specifies the percentage of trains that will be dispatched no earlier than the next day. The number of trains waiting to be dispatched each day is shown by the accumulator "train waiting Altynkol". Using the constant "init train Altynkol number", the user specifies the number of waiting trains at the time the model is launched. The parameter "train output daily limit Altynkol" is specified as the maximum number of trains that can be dispatched from the Altynkol station for one day.

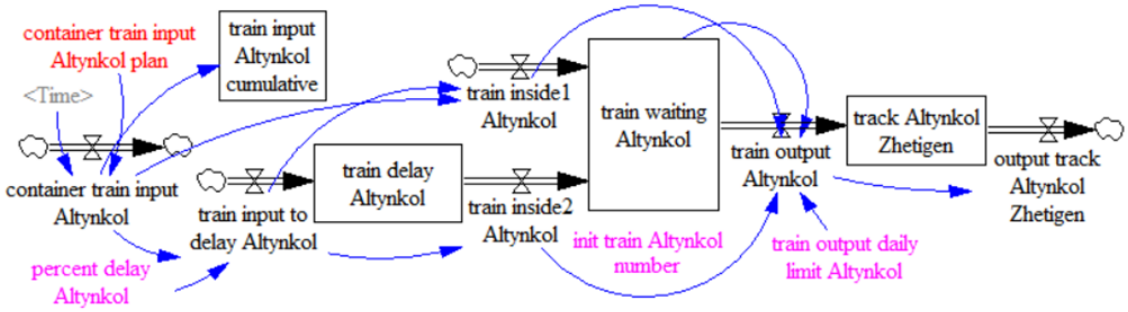


Figure 5. Structure of the Station Altynkol module

5.3. Presentation of simulation results

All model variables, which are both initial data and simulation results, can be displayed as flow charts or storage contents after the model is launched. In total, the model can be used to analyse more than 70 different processes in the transport corridor, for which data for charts is automatically created.

Table 1 illustrates, as an example, the results of modelling a scenario where there is no obvious accumulation of either trains or ferries at the ports. In modelling this scenario, it was assumed that the capacity of the stations is unlimited and that trains cover all sections of the railway line in the shortest possible time, i.e., in 1 or 3 days depending on the length of the section. As initialization data for the model, 16 trains that are already in transit on different sections of the route were specified. Two trains were already waiting to be loaded at the ports. Ferries appeared in the system only on the first day of the process.

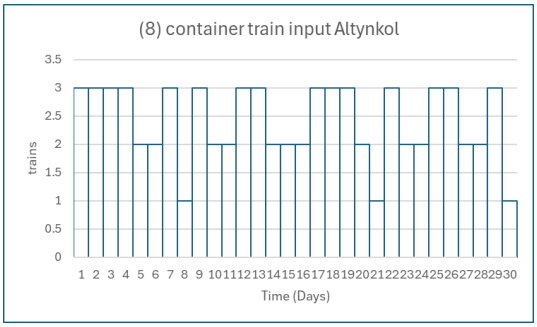
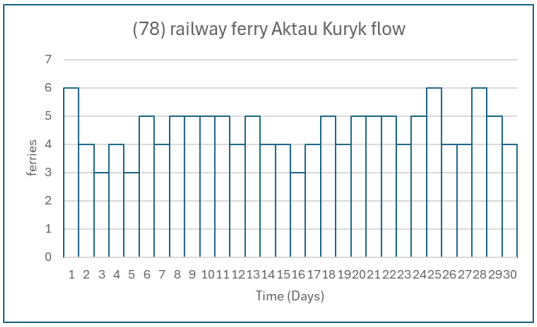
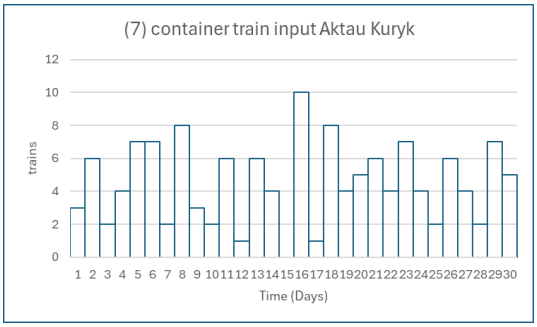
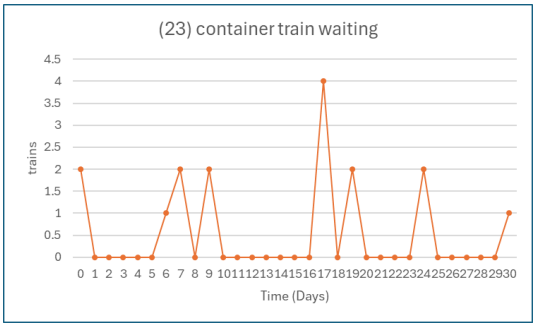
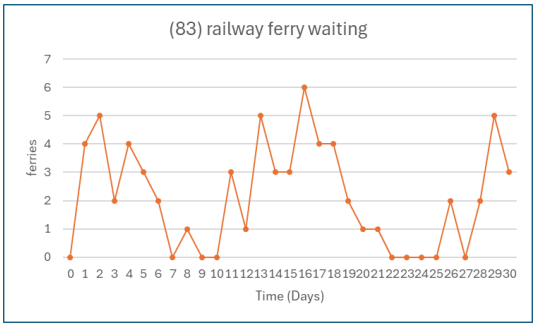
6. Conclusion

The classification of models presented in Figure 1 indicates that when formulating the properties of a model, it is necessary not only to classify it as one of the known process modelling paradigms, but also to clearly define the types of its "time/state" features. As a result of analysing the diversity of applied model types, the following list of actually existing process modelling paradigms can be compiled: Discrete-Event, Agent-Based, System Dynamics, Dynamic Systems, Discrete Time and Discrete Rate.

The developed model serves as evidence that a model cannot always be unambiguously attributed to one of the standard modelling paradigms. The model shows the movement of material flow objects, which is most often modelled using the Discrete-Event and Agent-Based paradigms. Although a package based on the System Dynamics paradigm was used to build the model, many of the model's properties make it different from conventional continuous models of this type. For the developed model, its belonging to the Discrete Time paradigm was determined. In the absence of clear restrictions on the properties of such models, a free choice was made of the level of detail for displaying the processes occurring in a real system, as a result of which the model acquired the properties of a mesoscopic model.

The applied simulation technique using the Vensim PLE package has a few advantages. Very simple and concise means are used to describe the properties of the model elements and the processes occurring in the model (see the description of the Station Altynkol module). The model generates data for constructing more than 70 process diagrams without the programming costs that would be mandatory when using other simulation tools.

Table 1. Fragments of initial data and simulation results of one scenario

	*Initial data:* Flow "Trains arriving at Altynkol station" During the day, one, two or three trains arrive at Altynkol station. All trains are directed to Saksaul station. The train arrival schedule is set by the user.
	*Initial data:* Flow "Ferries arriving at Aktau/Kuryk ports" Three to six ferries arrive at ports during the day. The ferry arrival schedule is set by the user.
	*Simulation result:* Flow "Trains arriving at Aktau/Kuryk ports" As a result of adding up the three input flows, from zero to 10 trains per day arrive at Aktau/Kuryk ports. Only on day number 15 were there no trains. On day number 16, 10 trains arrived once.
	*Simulation result:* Storage "Trains waiting for loading at Aktau/Kuryk ports" The number of trains in the waiting state on normal days is observed within the range from 0 to 2. After the peak arrival of 10 trains per day with number 16, 4 trains were in the waiting state during the day with number 17.
	*Simulation result:* Storage "Ferries waiting to load in Aktau/Kuryk ports" The number of ferries in the waiting state is observed within the range from 0 to 6. The wide range of values is explained by the interaction of two flows: the flow of trains arriving at the ports and the flow of arriving ferries.

References

1. Borshchev, A. (2013) *The big book of simulation modelling. Multimethod modelling with AnyLogic 6*. AnyLogic North America.
2. Buss, A., Al Rowaei, A. (2010) A comparison of the accuracy of discrete event and discrete time. In: *Proceedings of the 2010 Winter Simulation Conference*, Baltimore, December 2010. IEEE, 1468–1477.
3. Damiron, C., Krahl, D. (2014) A global approach for discrete rate simulation. In: *Proceedings of the 2014 Winter Simulation Conference*, Savannah, December 2014. IEEE, 2966–2977.
4. Elizandro, D., Taha, H. (2007) *Simulation of industrial systems. Discrete event simulation using Excel VBA.* Auerbach Publications.
5. Hennies, T., Reggelin, T., Tolujew, J., Piccut, P.-A. (2014) Mesoscopic supply chain simulation. *Journal of Computational Science*, 5(3), 463–470. doi:10.1016/j.jocs.2013.08.004.
6. Love, D. (2017) Dynamic simulation on a spreadsheet as a tool for evaluating options for mixed juice flow control. In: *Proceedings of the 90th South African Congress of the South African Sugar Technologists' Association (SASTA)*, 90, 366–381. Available: https://www.cabidigitallibrary.org/doi/pdf/10.5555/20183262241.
7. Morrison, A., Diesmann, M. (2007) Maintaining causality in discrete time neuronal network simulations. *Lectures in Supercomputational Neurosciences. Understanding Complex Systems*. Springer, Berlin, Heidelberg. Available: https://doi.org/10.1007/978-3-540-73159-7_10.
8. Phelps, R. A., Parsons, D. J., Siprelle, A. J. (2002) Non-item based discrete-event simulation tools. In: *Proceedings of the 2002 Winter Simulation Conference*, San Diego, December 2002. IEEE, 182–186.
9. Reggelin, T., Tolujew J. (2011) A mesoscopic approach to modelling and simulation of logistics processes. In: *Proceedings of the 2011 Winter Simulation Conference*, Phoenix, December 2011. IEEE, pp. 1513–1523.
10. Reggelin, T., Lang, S., Schauf, Ch. (2020) Mesoscopic discrete-rate-based simulation models for production and logistics planning. *Journal of Simulation*, 16(5), 448–457. https://doi.org/10.1080/17477778.2020.1841575.
11. Riezebos, S. (2016) *Discrete-time simulation and optimization of multi-echelon distribution systems*. Master thesis, Eindhoven University of Technology. Available: https://dc.wtb.tue.nl/lefeber/do_download_pdf.php?id=169.
12. Savrasovs, M. (2012) Traffic flow simulation on discrete rate approach base. *Transport and Telecommunication*, 13(2), 167–173. doi:10.2478/v10244-012-0014-8.
13. Seila, A. F. (2003) Spreadsheet simulation. In: *Proceedings of the 2003 Winter Simulation Conference*, Monterey, December 2006. IEEE, 25–30.
14. Sun, B., Appiah, J., Park, B. B. (2020) Practical guidance for using mesoscopic simulation tools. *Transportation Research Procedia*, 48, 764–776. doi:10.1016/j.trpro.2020.08.078.
15. Tang, J., Leu, G., Abbass, H. A. (2020) *Simulation and computational red teaming for problem solving*. Hoboken: IEEE Press. doi:10.1002/9781119527183.
16. Toney, J., Jayakumar, A. (2022) *MATLAB programming for engineering applications*. The Ohio State University.
17. Vensim. (2024) *Vensim® Personal Learning Edition.* Available: https://vensim.com/vensim-personal-learning-edition/.
18. World Bank. (2023) *Middle trade and transport corridor: Policies and investments to triple freight volumes and halve travel time by 2030*. Washington, DC: World Bank. Available: https://www.worldbank.org/en/region/eca/publication/middle-trade-and-transport-corridor.
19. Zeigler, B. P., Kofman, E. (2018) *Theory of modelling and simulation*. Third Edition, Academic Press.