ROAD TRAFFIC MODELLING BASED ON THE HYBRID MODELLING TOOL ANYLOGIC

Gabriel **Benčat** [1,*] , Aleš **Janota** [2]

[1] University of Žilina, FEIT, Dept. of Control and Information Systems, Univerzitná 8215/1, 010 26 Žilina, Slovakia, email: bencat2@stud.uniza.sk,
https://orcid.org/0000-0002-2750-9179

[2] University of Žilina, FEIT, Dept. of Control and Information Systems, Univerzitná 8215/1, 010 26 Žilina, Slovakia, email: ales.janota@uniza.sk,
https://orcid.org/0000-0003-2132-1295

* Corresponding author

Abstract – The increasing complexity of modelled systems opens up questions of combining several modelling approaches in order to achieve higher efficiency and the required clarity of the simulations performed. For this reason, in addition to standard application-oriented modelling tools, we can see the growing popularity of the so-called hybrid or multimethod modelling approaches. A typical representative of such a modelling environment is the AnyLogic general-purpose modelling tool. This article describes the authors' experience with this tool in solving a selected problem in the field of modelling transport processes in urban areas. Using the example of the uncontrolled selected intersection, the procedure of creating a model with a more detailed analysis of its selected parts is discussed. The model based on real conditions is elaborated and compared with the model containing the fictitious traffic light control. Comparison is based on measuring the time that vehicles moving in the critical direction spend in the model. Both discussed models are stored in the cloud and may be run and study free.

Key words – AnyLogic, model, simulation, traffic, agent

JEL Classification – L9, 018, R41

INTRODUCTION

Computers allow researchers to simulate how different types of devices or processes work. To make this possible, a set of assumptions are to be defined in the form of logical or mathematical relationships that make up the model. Models being simple enough make possible to apply analytical (mathematical) approaches; however, most real-world systems are too complex and may be studied by means of simulation only. The domain of road transportation and traffic operation is not an exception. For a comprehensive and state-of-the-art treatment of all the important aspects of a simulation study, see [1]. A comprehensive, in-depth, and state-of-the-art summary of the important aspects of transportation analysis and modelling and simulation is provided in [2].

Various kinds of traffic models are being developed to help researchers answer questions such as: „How will the traffic evolve with the given

initial state and defined behavior of the participants in the real (or planned) transport network?"; „How to present the properties of road networks?"; „Under what conditions can traffic congestion be expected to arise and how to avoid it?"; „How best to manage incidents?"; „How to analyze traffic conditions so that the optimal way of traffic control can be applied?"; "How to perform what-if decision making?"; „How to test proposed solutions?"; „How to train traffic-oriented system users?"; and many others. Selected problems related to modelling tools in service of development of traffic engineering are presented in [3]. Simulation packages may be classified to general-purpose simulation packages (intended for any application), and an application-oriented simulation packages (intended for a certain class of applications).

As the performance of computer tools grows, so does the importance of visualizations and animations. In addition to traditional analytical models, a number of models using advanced

visualization solutions are created, e.g. with virtual reality and data-driven support. Progress is visible in all categories of model-based simulation methods - microscopic, mesoscopic and macroscopic ones, classified based on the levels of simulated details. An historical overview of the development of traffic flow models in the form of a model tree is available in [4]. Advantages and disadvantages of macroscopic models together with new insights are reviewed in [5]. A traffic simulation study usually consists of eight particular tasks, from formulation of the aims and scope of the study to documentation, see [6].

A comprehensive review on the state-of-the-art techniques for traffic simulation and animation is available in [7], with special attention paid to autonomous driving applications. Authors of the survey [8] are interested in real-time computer network applications. The source [9] presents a review and critical analysis of the mathematical literature on modelling of vehicular traffic and crowd phenomena, similarly a review of the mathematical models for traffic flow is available in [10]. Special attention should be paid to signal control methods, which can be implemented in models. The survey [11] covers various transportation approaches widely applied in traffic signal control. Another problem of many studies is model validity. The source [12] provides review on how traffic simulation models are being used, and what procedures are used for their calibration and validation. Some authors prefer more formal approaches, e.g. [13] present cellular and timed automata to simulate and formally verify urban road traffic. With the recent evolutions in Cloud Computing and Software-as-a-Service (SaaS), there is a new paradigm where simulation software is used in the form of services [14]. That approach makes possible to solve the problem of various methods integration and interoperability. One of the recent studies comparing various traffic simulation software tools is [14], using 8 basic criteria; a systematic literature review of existing vehicular traffic flow simulators is in [15-16] focuses on the representation of road traffic for the simulation of highway vehicular networks based on V2V communication technologies.

The paper has been elaborated with support of the KEGA Grant Agency.[1]

The aim of the paper is to present the experience with the creation of a traffic model of a road intersection in a hybrid environment primarily not intended for this application domain. The reason is to verify the usability and advantage of combining several modeling approaches in one model.

I. ANYLOGIC

Each modelling approach has its advantages and disadvantages. Therefore, the so-called hybrid or multimethod modelling tools are being developed that allow integrating multiple modelling approaches and making the most of each. This makes it possible to achieve higher modelling and simulation efficiency. One such tool is AnyLogic (www.anylogic.com) – the general-purpose simulation tool for creating dynamic environments [17]. It integrates three basic modelling principles, each of them recommended under different conditions:
− System dynamics modelling (if there is only information about global dependencies);
− Discrete event modelling (if a system is easily described as a process);
− Agent-based modelling (if there are many independent objects).

System dynamics approaches system modelling at a high level of abstraction, i.e. it ignores small details and creates a general representation of a complex system. The basic concepts of this approach are feedback loops. Mathematically, the model of non-linear dependencies in the real-world maps to a system of differential equations that are solved numerically. AnyLogic offers the object-oriented approach and enables complex models to be defined in a hierarchical manner with objects only exposing interface variables as inputs and outputs.

Discrete event modelling models processes at a medium level of abstraction when physical details are not represented. It is appropriate when the system under analysis can be described as a sequence of operations. Classic discrete event tools work with passive entities having certain attributes only.

Agent based modelling focuses on the individual active components of a system. Agents may have their individual behaviour and state changes and connections between them may be established. Agent-based models are usable at all levels. The global dynamics of the system then emerge from the interactions of the many individual behaviours.

Complex systems have different aspects, so the

approaches built-in AnyLogic can be combined appropriately. In addition, artificial intelligence (AI) methods such as neural network training, synthetic data generation or AI testing may be implemented. Created models may be stored in the Cloud and shared as needed.

For the sake of illustration, let us mention some of AnyLogic-based research studies related to road transport domain: the source [18] provides a literature review focusing on the MASs and their applications in traffic and transportation; [19] investigates routing algorithms for automated highway systems with AnyLogic used to simulate the behavior of individual cars; a pre-built AnyLogic Urban Dynamics model with considering population, business, housing, and transport infrastructure is employed to simulate the private EV electricity demand per hour in [20]; in [21] AnyLogic comprises agent-based and discrete-event techniques to ensure the balance of interests of transport companies trying to find compromise between cost of operation and quality of service. AnyLogic users can draw from a variety of sources, e.g. [22-24].

II. MODEL DESIGN

REQUIREMENTS SPECIFICATION

The intention was to create a model for one specific road intersection located near the university campus in Žilina, Slovakia (Fig. 1.). Drivers are currently crossing the intersection based on the application of the basic road traffic rules. The selected intersection connects the two side streets Karpatská and Veľký diel with the main street Vysokoškolákov. Drivers coming from side roads are obliged to give priority to those driving along the main road. A special group of transport means are buses of urban mass transport, which also have

a parking space there (at a one-way part of Veľký diel communication). Along the intersection there are sidewalks built for pedestrians. The first requirement is to create a model that would make it possible to simulate current traffic conditions at this intersection and measure the times needed to turn left from Veľký diel Street to the left (north) which nowadays seems a critical route, especially in rush hours. Vehicles enter and exit the selected communication network to/from all directions. Traffic intensities are to be estimated based on approximate observations. In this phase of modelling, the emphasis is more on creating a frame functional model (its procedural nature) than on obtaining as accurate statistic data as possible. The second requirement is to consider a traffic light control system virtually installed there and create fictitious conditions for experimenting with the model (a proper comparison of selected functionality).

TRAFFIC OPERATION MODEL WITHOUT TRAFFIC LIGHT CONTROL (REAL CONDITIONS)

The model was created using the AnyLogic tool (a free Personal Learning Edition) with the support of a road traffic library. A detailed systematic description of model design procedure is available in [25]. In order to summarize the implemented steps, we briefly describe a sequence of individual modelling actions, some of which supplemented by comments and described in more details. After establishing a new model, we implemented a Google-based satellite map as a working background. One of the important things to do was adjusting the scale on the map accordingly. Then we could start drawing road communications. The next step consisted in adding agents. It is always possible to select some of the offered options:

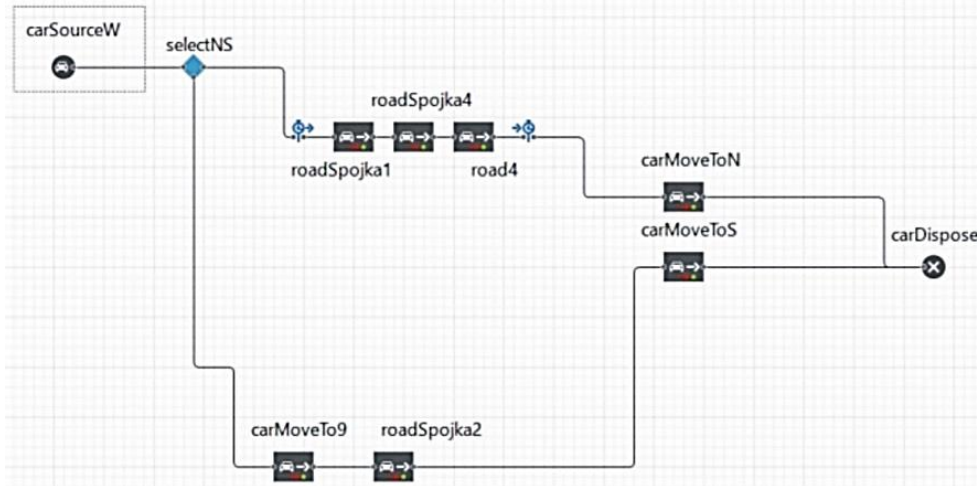**Fig. 1. Topology of the modelled intersection (Source: Google Maps)**

Fig. 2. A part of the process flowchart (**Source: own work**)

– *Population of agents*: creates several agents of the same housing in the same environment as the current agent;
– *A single agent*: creates a single agent that will always exist within the current agent;
– *Agent type only*: only the agent type will be created.

When creating a vehicle or pedestrian agent, we used the option of creating several agents, for the buildings we used the single agent option. One of the agents called the *Carforgraphs* agent is different from others - it is used to measure the time vehicles of a certain type spent in the simulation environment. This type of agent subtracts the start time from the exit time when a car agent is leaving the set point, and writes it into the variable used to plot a histogram for us.

Then we started creating logics of the traffic process represented by the process flowchart. In the AnyLogic, flowcharts are created by adding blocks from the library palette (in our case blocks of the road traffic and pedestrian libraries) to the workspace, where they are then merged. Each block defines a specific operation to be performed. When moving blocks and placing them side by side, it is essential to ensure that all blocks are properly interconnected. Each flowchart starts with a starting point or source of elements (for a road traffic library, this is the *CarSource* block type).

To illustrate the way of how process flow charts are created, see Figure 2. It starts in the left top corner with the element *carSourceW* representing a transport means entering the model from the West (left side). Then it follows the route along

Veľký diel Street and turns to the left (i.e. moves to the North) or to the right (i.e. moves to the South). It is necessary to set properties such as the intensity of occurrence, the type of agent to be generated, the initial or preferred speed of the agent, and other options. It is also important to set the possible route (where the agent should appear - here on the road named *roadW*). As a type of vehicle (here *Carforgraphs* agent) we used a car type with a pink colour. Since our transport means called *CarSourceW* must make decision whether to travel upwards to the North or downwards to the South we use the *selectOutput* block type, named *selectNS*, to branch out the flowchart process. Which branch the agent decides to go depends on the probability we have chosen in the block. Since in reality more vehicles are heading the North (to the city centre), we set probability to 0.7. When the agent leaves *roadW*, it must continue to *roadSpojka1* whether it is heading the North or the South. We defined its behaviour using the *CarMoveTo* block. Care must be taken in which direction to leave the road and in which direction to go. Agents turning to the North must get to roads *roadSpojka1*, *road4*, *roadN* up to the end block *carDispose*. Agents turning to the South will go to the roads *roadSpojka1*, *roadSpojka2*, *roadS* to the end block *carDispose*. By connecting the blocks, the displayed flowchart is completed. A similar logic was used when creating other traffic flows, only with different parameters and different agents. The complete process flowchart is shown below (a partial flowchart displayed in Figure 2 can be identified there as well).
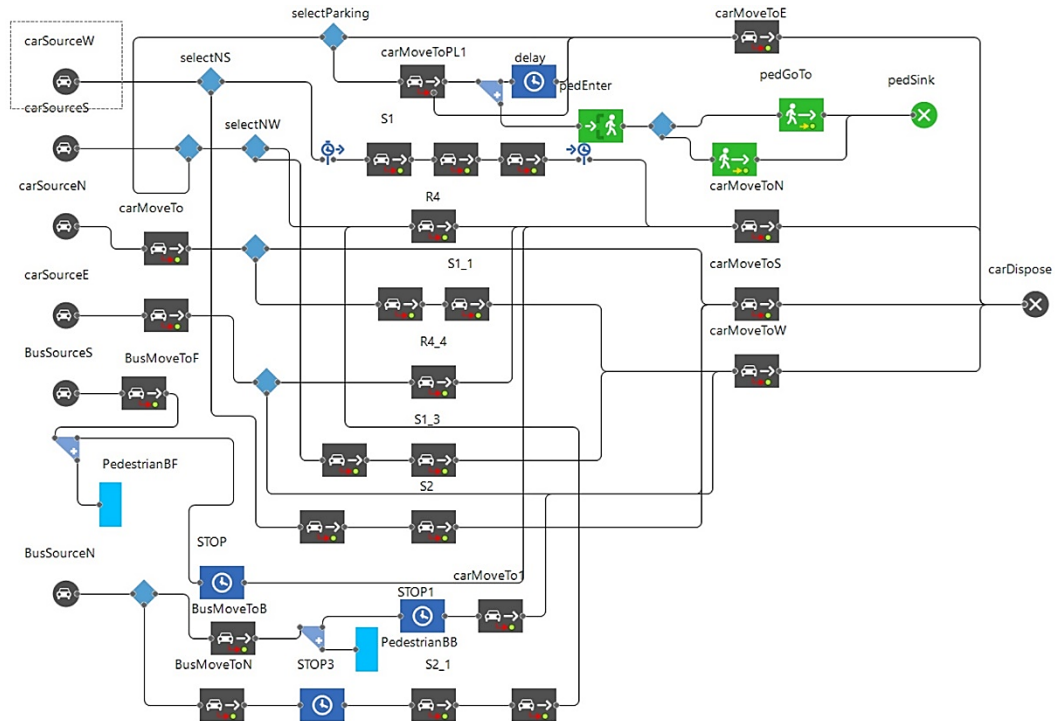
Fig. 3. The process flowchart (**Source: own work**)

Sometimes, when we create a flowchart, we can reach the point where a clarity is lost or there is no or little space to display a certain element. In such a case, it is advisable to create an own flowchart block - we used it to simulate passengers getting off the mass urban transport bus. The space in the flowchart was already limited (too densely drawn), so we created the *PedestrianBF* and *PedestrianBB* blocks for greater clarity. Those blocks are represented by blue rectangles (Fig. 3.) as instances of the new agent type.

Adding 3D windows enables us to run 3D simulations; multiple windows may display specific parts of the scene. Certain parts of the transport infrastructure can be secured by inserting a camera into the workspace so that it captures the required sections. A 3D window can utilize several types of navigation (e.g. *Full* manipulation, *Rotation only*, *Limited to Z above 0* for horizontal manipulation only, *None*). In the case of large and complex models, agents usually contain a large number of elements, which can result in important parameters not being displayed. For certain models, it is impossible to place all of these elements in the area of the display frame that is displayed during the simulation. AnyLogic offers a special element for solving this problem and that is the field of view. Using this element, we can mark some specific areas in the simulated program, which contain, for example, logically separated groups of elements or other parts of the program. After defining such view areas, we can easily switch between them in the runtime model using special navigation tools. We can define several separate view areas (one for each viewing area), switch between these areas at runtime, and examine how the different areas work.

3D animation provides the most realistic and natural way to visualize the simulated process. Standard shapes are usually used to draw simple objects, such as roads, walls, boxes, and so on. More complicated objects such as people, carts, planes are not common in AnyLogic. They must be imported from the outside environment using a special 3D object element. AnyLogic supports the import of 3D models that are stored in Collada (.dae) files. In our model, we used only a few such objects to reduce the complexity and demandingness of the simulation. For example, the ground plan was created using a rectangle, where

we used grass as the colour fill.

To illustrate visualization abilities of the programming environment, we can show visualization of the public bus stop (Fig. 4). In our model there are two bus stops - the first located in the direction from the North to the South and the second in the opposite direction. To simplify the process flowchart, we created our own flowchart block. When the bus arrives at the bus stop, the process branches using a split block and some other blocks are used (pedestrian entrance; the destination, where to go; the block of extinction). The process flowchart for pedestrians is created on

the same principle as mentioned before.

We also created a parking place with seven parking lots on the *roadE* heading the East (Fig. 5). To make this possible, we had to branch out the flowchart using the *SelectOutput* block type and instruct the vehicle to head the marked parking lot. Using the delay block, we set a parking time. The *CarMoveTo* block type contains one input and two outputs (in case of only one output, if the parking place was occupied, the output would not be available and the agent would have nowhere to get). In this way, the agent can continue driving if it intends to park but all parking slots are occupied.

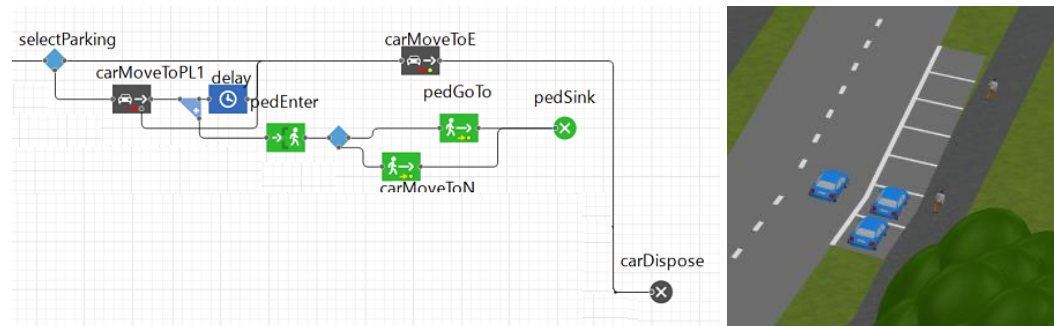**Fig. 4. The 3D model of the bus stop (Source: own work)**

Fig. 5. The process flowchart and the 3D model of the parking place (Source: own work)

There are more places in the simulated world where agent movements must specially be controlled to get closer to the real traffic operation. For their control we used another modelling approach supported by AnyLogic – the Statecharts formalism. Red circles in Figure 6 (top) indicate areas controlled just by the statecharts. Let us have more details on how movement of pedestrians and cars can be modelled. The area of pedestrian crossing is divided into four zones (1 - *areaInput*, 2 - *areaI*, 3 - *areaO*, 4 - *areaInput2*) and 2 stop lines for both car movement directions (*stopLinePPF*,

stopLinePPB). This place slows down the means of transport in smooth driving and subsequent turning. To analyse all the states and the transitions between them, we created the state machine (for the purpose of this paper, the simplified state machine depicted in Figure 7 was drawn in Visual Paradigm). To implement statechart blocks, we defined auxiliary variables and blocks defining rules of pedestrian behaviour in a certain area. In state S1, no pedestrian passes through the pedestrian crossing and an enabling signal implemented by the respective internal signal is transmitted.
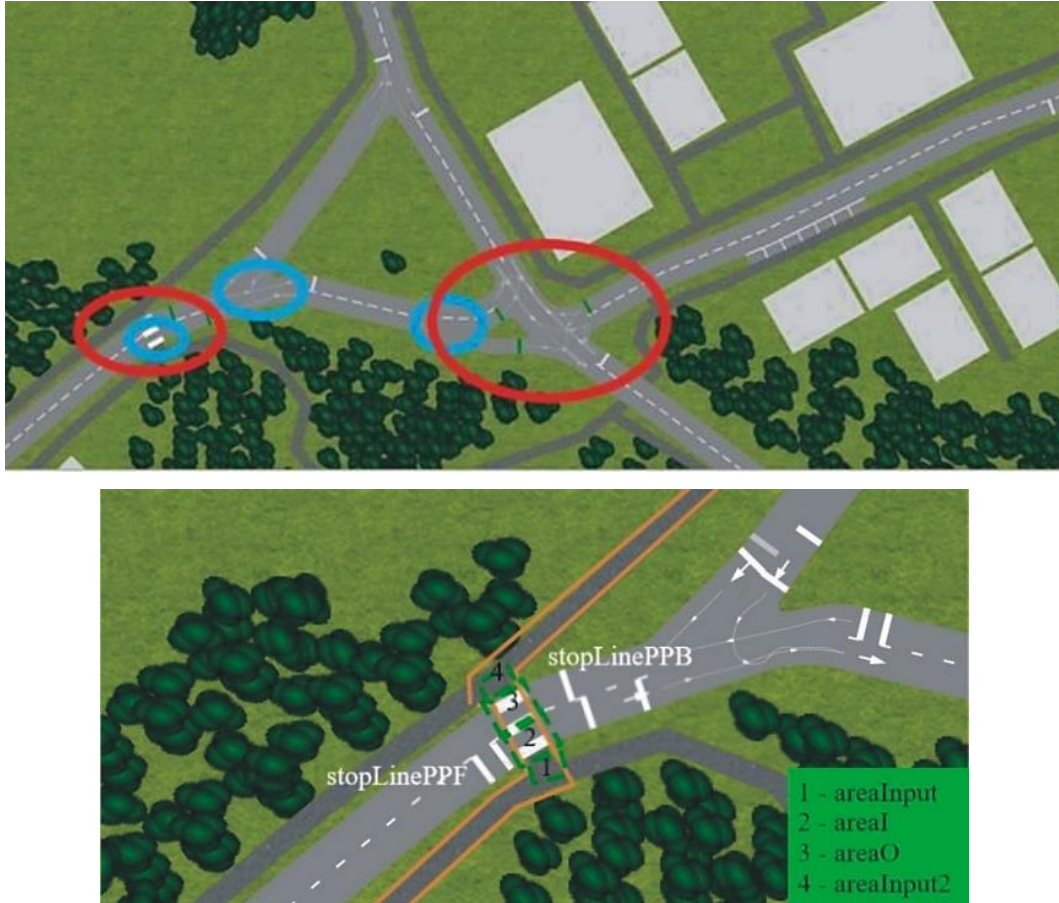
Fig. 6. The areas controlled by the statecharts and a detailed look at the pedestrian crossing (Source: own work)

The state repeats cyclically every 500 ms. If a pedestrian enters *areaInput* or *areaI*, the signal is interrupted and the statechart gets to state S2.1 which is cyclically tested every 300 ms. The value of the auxiliary variables increases by one (not indicated here) according to the input area. This signals the pedestrian's entry into the section and sends a forbidding signal to *stopLinePPF*. After the pedestrian exits the first area, the value of the auxiliary variable decreases by 1. As he/she exits the first area, he/she immediately enters the second area, finds already on the road and the statechart gets to state S3; conditions are cyclically tested every 100 ms. In this state, it sends a stop label to *stopLinePPF* and *stopLinePPB*. After exiting the second area, the pedestrian passes to the third

area; the statechart remains in the same state. When a pedestrian exits the third area and enters the fourth, he/she is already out of the way. Then a proceed aspect is sent to *stopLinePPF*, but *stopLinePPB* is still in stop position since the pedestrian is still close to the road pavement. Analogous behaviour occurs when a pedestrian moves in the opposite direction. Similarly, we implemented the model of the intersection itself. From all considered processes, this model was the most complicated. We defined a control algorithm (again based on the state machine), which, based on the description of the road traffic rules, ensured conflict-free crossing of the intersection (see Fig. 8.). The individual agent knows only the basic traffic rules; the more complex ones needed to be

defined. To control the movement of road users, we created auxiliary variables and added stopline blocks (Fig. 8.) that helped us determine the position of vehicles. Several other statecharts were created (Fig. 9.), e.g. *StateforBusHelp* making running of public buses fluent, or *StateforcarHelp*

checking the number of vehicles passed when splitting a road. Without control, the vehicles got stuck in front of *stopLine2* stacked on top of each other (due to the fact that the transition from one route to another was very short, the agent could not determine the distance and collisions occurred).
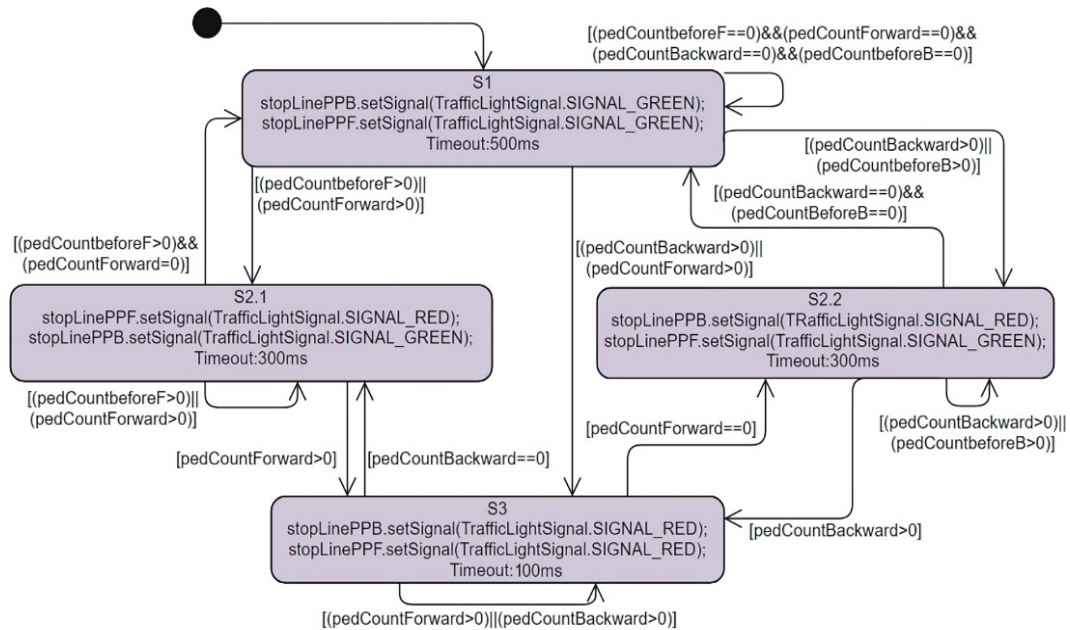

Fig. 7. The state machine describing behaviour at the pedestrian crossing (Source: own work)

Fig. 8. The intersection configuration in the model without traffic light control (Source: own work)
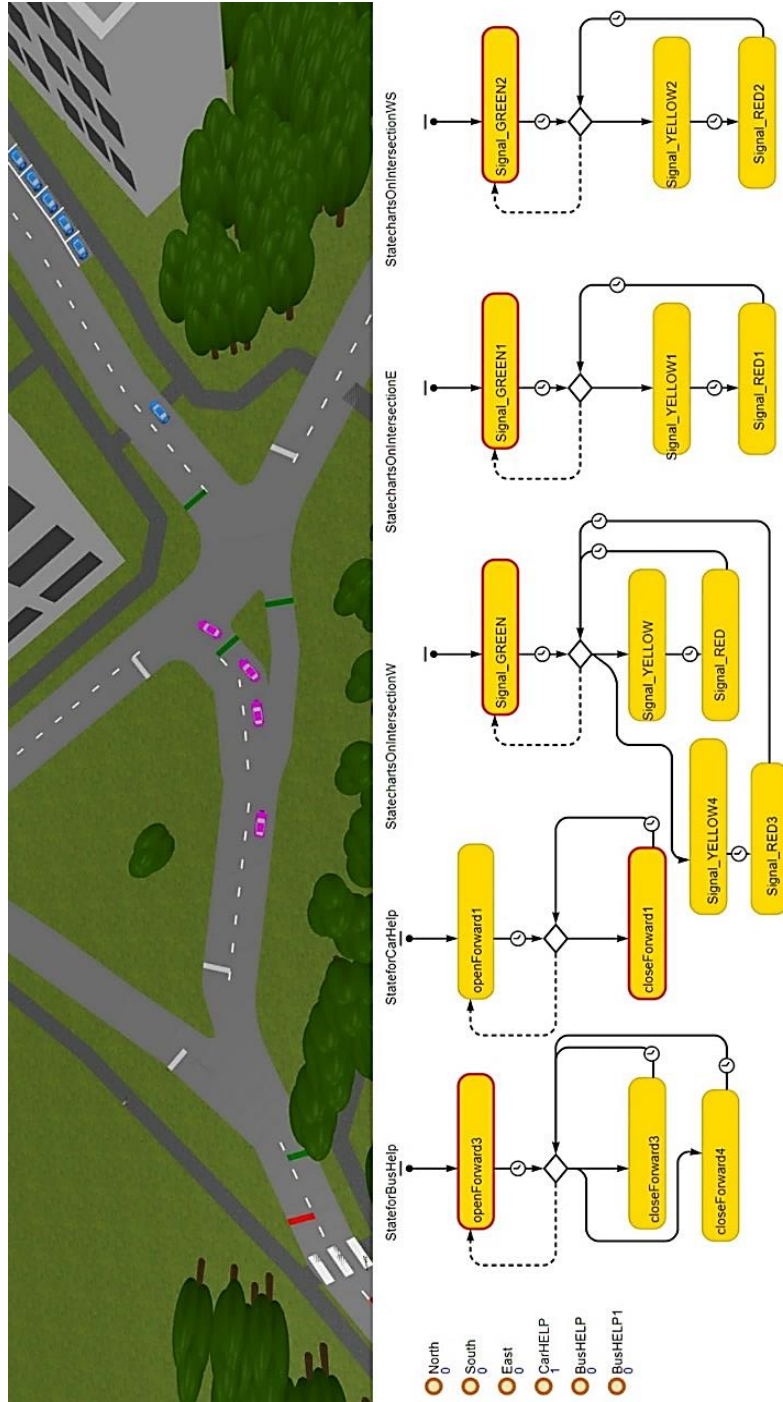
Fig. 9. Illustration of animated statecharts in the model without traffic light control (Source: own work)

The model can be run in either 2D or 3D visualisation (Fig. 10.).

Fig. 10. Two-dimensional and three-dimensional visualizations of the same scene (**Source: own work**)

TRAFFIC OPERATION MODEL WITH TRAFFIC LIGHT CONTROL (FICTITIOUS CONDITIONS)

The aim of the next steps was to supplement the previous model with a new functionality - traffic control at the intersection via a traffic light. Initial considerations assumed the use of the built-in *TrafficLight* element from the traffic operation library. When using it, it is possible to proceed in the standard way, where the control algorithm is formalized as a signal plan of the traffic controller (Fig. 11 top). The image comes from an older (transitional) version of the model and is therefore used here for illustrative purposes only. However, the obtained solution brought us certain problems - assigning vehicles into particular individual phases despite of being not at the section, significantly fluctuating traffic intensity in the *EastE* section (for most of the day, very weak) coming from the housing estate, etc.
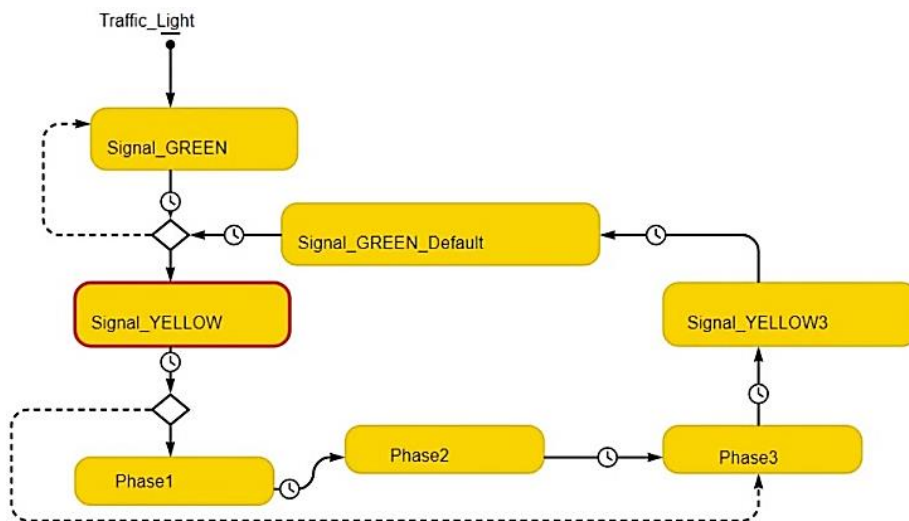
Fig. 11. Signal plan (top) and statechart (down) for the traffic light (Source: own work)

Therefore a custom solution of the control algorithm based on the own control algorithm was designed (Fig. 11 down). Thus both pictures contained in Figure 11 are not related in any way. The simplified state machine was re-drawn for the purpose of this paper in Visual Paradigm (Fig. 12.).
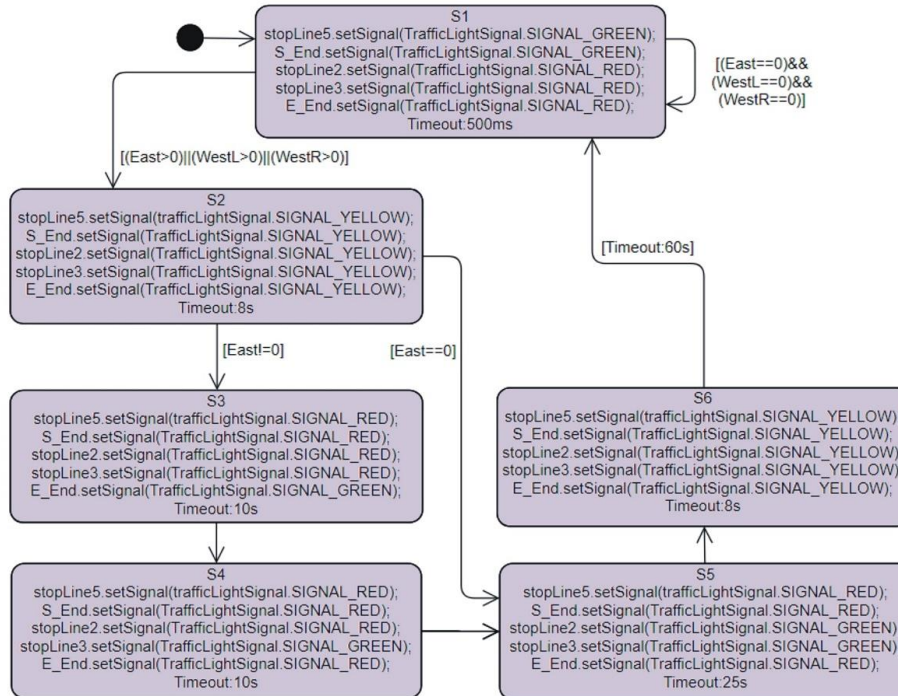
S1
stopLine5.setSignal(TrafficLightSignal.SIGNAL_GREEN);
S_End.setSignal(TrafficLightSignal.SIGNAL_GREEN);
stopLine2.setSignal(TrafficLightSignal.SIGNAL_RED);
stopLine3.setSignal(TrafficLightSignal.SIGNAL_RED);
E_End.setSignal(TrafficLightSignal.SIGNAL_RED);
Timeout:500ms

[(East==0)&&
(WestL==0)&&
(WestR==0)]

[(East>0)||(WestL>0)||(WestR>0)]

S2
stopLine5.setSignal(trafficLightSignal.SIGNAL_YELLOW);
S_End.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
stopLine2.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
stopLine3.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
E_End.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
Timeout:8s

[Timeout:60s]

[East!=0]   [East==0]

S3
stopLine5.setSignal(trafficLightSignal.SIGNAL_RED);
S_End.setSignal(TrafficLightSignal.SIGNAL_RED);
stopLine2.setSignal(TrafficLightSignal.SIGNAL_RED);
stopLine3.setSignal(TrafficLightSignal.SIGNAL_RED);
E_End.setSignal(TrafficLightSignal.SIGNAL_GREEN);
Timeout:10s

S6
stopLine5.setSignal(trafficLightSignal.SIGNAL_YELLOW);
S_End.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
stopLine2.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
stopLine3.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
E_End.setSignal(TrafficLightSignal.SIGNAL_YELLOW);
Timeout:8s

S4
stopLine5.setSignal(trafficLightSignal.SIGNAL_RED);
S_End.setSignal(TrafficLightSignal.SIGNAL_RED);
stopLine2.setSignal(TrafficLightSignal.SIGNAL_RED);
stopLine3.setSignal(TrafficLightSignal.SIGNAL_GREEN);
E_End.setSignal(TrafficLightSignal.SIGNAL_RED);
Timeout:10s

S5
stopLine5.setSignal(trafficLightSignal.SIGNAL_RED);
S_End.setSignal(TrafficLightSignal.SIGNAL_RED);
stopLine2.setSignal(TrafficLightSignal.SIGNAL_GREEN);
stopLine3.setSignal(TrafficLightSignal.SIGNAL_GREEN);
E_End.setSignal(TrafficLightSignal.SIGNAL_RED);
Timeout:25s

Fig. 12. The state machine describing adaptive traffic control logics (Source: own work)

EXPERIMENTS

In addition to monitoring the visual course of the simulation and 2D/3D animation of the dynamics of the model, our attention was also focused on the question of what and how to measure. There are several ways to measure the required quantity in simulated time. Figure 13 illustrates the histogram that visualizes statistics collected by a number of histogram data objects. The *x*-axis (representing time in system) always is scaled to fit all histogram data ranges. The histograms are also scaled along the *y*-axis so that the highest bar occupies the full height of the picture. The probability distribution function (PDF) bars, the cumulative distribution function (CDF) line and the mean location can be shown optionally. PDF is displayed as a number of vertical bars each corresponding to a particular interval with heights proportional to the density (or number) of data samples within the interval. The bars may be setup to look as a solid block or distinct bars. CDF is displayed as a polyline on top of PDF, and the mean value as a black vertical line placed at the corresponding *x*-axis value position [26]. That average value is also a good performance

measure indicator of traffic signal control. The histogram in Figure 13 corresponds to the following settings: the model without traffic light control; situation captured after ca 180s of real time run - with 2 time units per second corresponding to ca 360s real time long simulation; focused on vehicles coming from the West. The graph is related to the *Carforgraphs* agent, measuring times of coming and leaving the model (not used in the experiment described in Fig. 15.). The *y*-axis in Figure 13 shows the percentage of agents taking amount of time, e.g. the value 0.2 means 20%. Agent presence in particular locations in the modelled system can be measured using the *TimeMeasureStart* and *TimeMeasureEnd* blocks. They connect a pair of objects, where they measure the time that agents spend between them. In principle, there may be more than one *TimeMeasureEnd* block referring to several *TimeMeasureStart* blocks. The graphs shown in Figure 14 are based on the same settings as mentioned before. The "Time plot" graph (on the right) shows time from the start of model run up to reaching the final measurement time (*x*-axis); time spent between *TimeMeasureStart* and *TimeMeasureEnd* blocks is drawn on the *y*-axis.
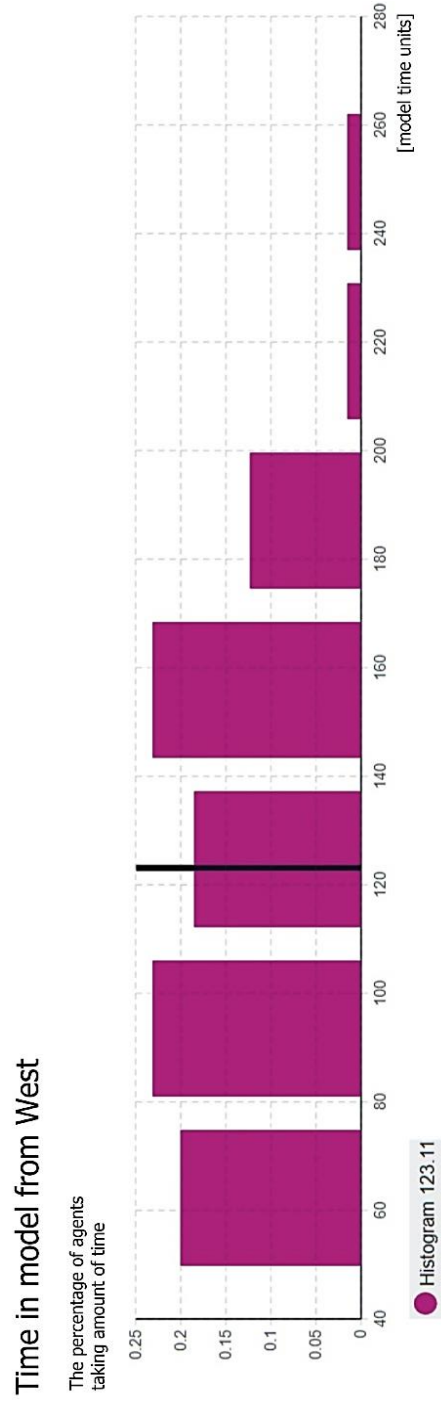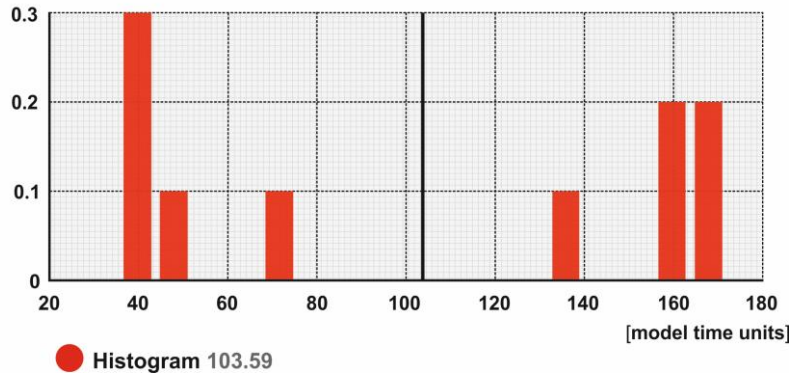
Fig. 13. Histogram-type diagram available in the AnyLogic tool (Source: own work)

Time to turn on North

The percentage of agents
taking amount of time

Histogram 103.59

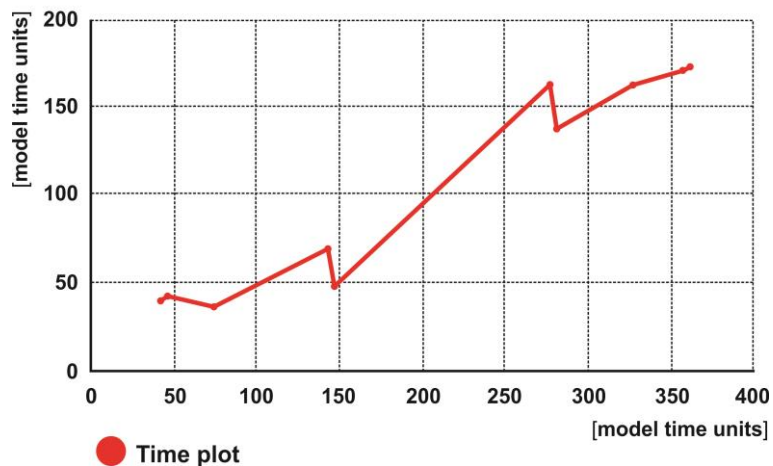## Time spent between *TimeMeasureStart/End* block

Time plot

Fig. 14. Histogram-type diagram available in the AnyLogic tool (Source: own work)

AnyLogic offers various ways to test the model or optimize certain parameters. For the purpose of this paper, we have chosen an experiment based on tracking the times needed to perform turn manoeuvre by vehicles coming from the West, for different traffic intensities on the *roadW* (Fig. 15.). By checking the item "log model execution" in the menu "*projects*" we were able to collect required statistics (using the *TimeMeasureStart* and *TimeMeasureEnd* blocks). The data was collected in the database in the *datasets_log* and *histograms_log* sections.

The results obtainable from the model are indicated in Table 1. The table shows nine measurements using various traffic intensities for variable numbers of agents (from 100 to 500 agents per hour) performed in both the model without and with light signal control. During simulation so called virtual speed was used (coping a certain time limit internally set in the personal edition). The table indicates that with traffic control we are able to reach better results – agents in the latter model were 1min 37s faster (i.e. they needed less time to go through the intersection). The complete tables to all experiments are available in [25]. The graphical representation of data from the table is depicted in Figure 16.
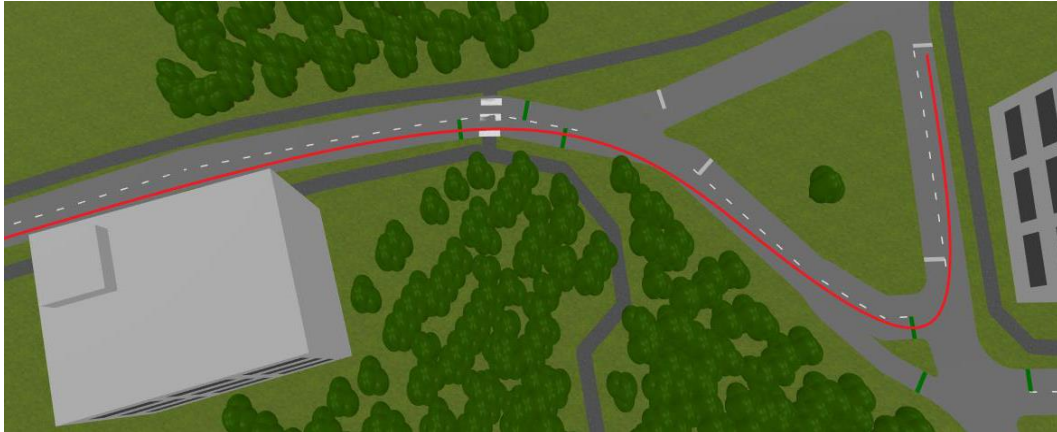
Fig. 15. The marked measured section (**Own work**)

Table 1. Samples of measurement results (**Source: own work**)

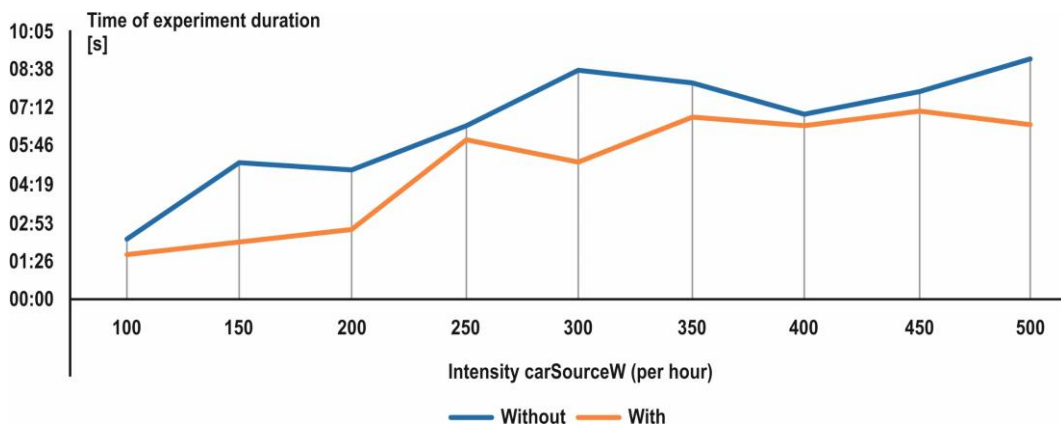|	Intensity carSourceW (per hour)									Average
	100	150	200	250	300	350	400	450	500	waiting
Without control	02:13	05:07	04:51	06:29	08:37	08:08	06:57	07:47	09:02	06:34
With control	01:41	02:00	02:38	06:01	05:10	06:53	06:34	07:03	06:33	04:57
Difference	00:32	03:06	02:13	00:28	03:27	01:15	00:23	00:43	02:28	01:37

Fig. 16. Graphical representation of selected experiment samples (**Source: own work**)

The readers are welcomed to use and study runs of the described models free. Both models are available in the AnyLogic cloud - the link to the cloud is https://cloud.anylogic.com/models, then search either for the model named *Model_NoTL* (for simulation without traffic light control) or for the model named *Model_TL* (for simulation with traffic light control), and run.

Conclusions

The paper presented the simulation solutions of a particular road intersection developed in the simulation general-purpose tool AnyLogic. Partial solutions were discussed in more detail; complete models are available to readers in cloud storage to run and study. The specific traffic intensities used in the model were only estimated, because the

primary goal of the work was to test the possibilities of the simulation tool and the advantage of combining several modelling approaches. Integration of agent-based, statechart and system dynamics seems to be very effective way of modelling that provides a wide range of creativity options for the designer. In principle, a problem may be solved in several ways - apart from built-in libraries it is always possible to elaborate own custom solutions. The framework models developed can be used for a more detailed future analysis of the traffic situation (using relevant data from actual traffic surveys) and proposals for future transport solutions (e.g. including simulation of potentially constructed roundabouts, adding additional pedestrian crossings at the bus stop, addressing vehicles with priority rights of travel, etc.).

ABBREVIATIONS

1. **2D** - Two-dimensional
2. **3D** - Three-dimensional
3. **AI** - Artificial Intelligence;
4. **CDF** - Cumulative Distribution Function
5. **MAS** - Multi-Agent System
6. **PDF** - Probability Distribution Function
7. **SAAS** - Software-as-a-Service;
8. **V2V** - Vehicle-to-Vehicle.

MODELOWANIE RUCHU DROGOWEGO W OPARCIU O NARZĘDZIE MODELOWANIA HYBRYDOWEGO ANYLOGIC

Rosnąca złożoność modelowanych systemów otwiera pytania o połączenie kilku podejść do modelowania w celu osiągnięcia wyższej wydajności i wymaganej przejrzystości przeprowadzanych symulacji. Z tego powodu, oprócz standardowych narzędzi do modelowania zorientowanego na aplikację, możemy zaobserwować rosnącą popularność tak zwanych podejść do modelowania hybrydowego lub multimetodowego. Typowym przedstawicielem takiego środowiska modelowania jest narzędzie do modelowania ogólnego AnyLogic. W tym artykule opisano doświadczenia autorów z tym narzędziem w rozwiązywaniu wybranego problemu w zakresie modelowania procesów transportowych na obszarach miejskich. Na przykładzie niekontrolowanego wybranego skrzyżowania omówiono procedurę tworzenia modelu z bardziej szczegółową analizą jego wybranych części. Model oparty na rzeczywistych warunkach jest opracowany i porównany z modelem zawierającym fikcyjną kontrolę sygnalizacji świetlnej. Porównanie opiera się na pomiarze czasu, w jakim pojazdy poruszają się w krytycznym kierunku w modelu. Oba omówione modele są przechowywane w chmurze i mogą być uruchamiane i badane bezpłatnie.

REFERENCES

[1] Law A.M. (2014) Simulation Modeling and Analysis, 5th Edition. McGraw-Hill Education, ISBN 978-0-07-340132-4

[2] Mőller D.P.F. (2014) Introduction to Transportation Analysis, Modeling and Simulation. Computational Foundations and Multimodal Applications. Springer, ISBN 978-1-4471-5636-9

[3] Directions of Development of Transport Networks and Traffic Engineering. Lecture Notes in Networks and Systems (2019) Macioszek E. and Sierpiński G. (Eds.), Springer Cham, ISBN 978-3-319-98614-2

[4] van Wageningen-Kessels F., van Lint H., Vuik K., Hoogendoorn S. (2015) "Genealogy of traffic flow models". EURO Journal on Transportation and Logistics. Vol 4, Issue 4. pp 445-473.
https://doi.org/10.1007/s13676-014-0045-5

[5] Zeng Y., Ran B. (2019) "Review and New Insights of Macroscopic Model for Road Vehicle Traffic Flow". 4th International Conference on Electromechanical Control Technology and Transportation (ICECTT), Guilin, China. pp 282-285.
doi: 10.1109/ICECTT.2019.00071

[6] Olstam J., Tapani A. (2011) "A Review of Guidelines for Applying Traffic Simulation to Level-of-service Analysis". Procedia Social and Behavioral Sciences. Vol 16. pp 771-780.
doi:10.1016/j.sbspro.2011.04.496

[7] Qianwen Chao, Huikun Bi, Weizi Li, Tianlu Mao, Zhaoqi Wang, Ming C. Lin, Zhigang Deng (2020) "A Survey on Visual Traffic Simulation: Models, Evaluations, and Applications in Autonomous Driving". Computer Graphics Forum. Vol 39, Issue 1. pp 287-308.
https://doi.org/10.1111/cgf.13803

[8] Mohammed A.M., Agamy A.F. (2011) "A Survey on the Common Network Traffic Sources Models". International Journal of Computer Networks (IJCN). Vol 3, Issue 2. pp 103-115

[9] Bellmono N., Dogbe Ch. (2011) "On the Modeling of Traffic and Crowds: A Survey of Models, Speculations, and Perspectives". SIAM Review. Vol 53, Issue 3. pp 409-463.
https://doi.org/10.1137/090746677

[10] Tyagi V., Darbha S., Rajagopal K.R. (2009) "A review of the mathematical models for traffic flow". International Journal of Advances in Engineering Sciences and Applied Mathematics. Issue 1. pp 3-68

[11] Wei H., Zheng G., Gayah V., Li Z. (2020) "A Survey on Traffic Signal Control Methods". arXiv:1904.08117v3 (preprint). Vol 1, Issue 1. 32 pages. https://arxiv.org/pdf/1904.08117.pdf (access date: 05/05/2020)

[12] Brackstone M., Montanino M., Daamen W., Buisson Ch., Punzo V. (2012) "Use, Calibration, and Validation of Traffic Simulation Models in Practice: Results of Web-Based Survey". TRB 2012 - Transportation Research Board 91rd Annual Meeting. Washington D.C, United States. 12 pages. hal-01411970

[13] Valente E., Avram C., Machado J., Astilean A. (2018) "An Innovative Approach for Modelling Urban Road Traffic Using Timed Automata and Formal Methods". Journal of Advanced Transportation. Vol 2018, Article ID 6269526. 15 pages.
https://doi.org/10.1155/2018/6269526

[14] Ejercito P.M., Nebrija K.G.E., Feria R.P., Lara-Figueroa L.L. (2017) "Traffic simulation software review". 8th International Conference on Information, Intelligence, Systems & Applications (IISA), Larnaca, 2017. pp 1-4.
https://doi.org/10.1109/IISA.2017.8316415.

[15] Mubasher M.M., ul Qounain J.S.W. (2015) "Systematic literature review of vehicular traffic flow simulators". 2015 International Conference on Open Source Software Computing (OSSCOM), Amman. pp 1-6.
https://doi.org/10.1109/OSSCOM.2015.7372687

[16] Hilt B., Berbineau M., Vinel A., Pirovano A. (2017) "Highway Road Traffic Modeling for ITS Simulation", Networking Simulation for Intelligent Transportation Systems: High Mobile Wireless Nodes, Wiley. pp 165-184, ISBN 9781848218536

[17] Anylogic: Simulation Software for Every Business Challenge. https://www.anylogic.com/features (access date: 05.06.2020)

[18] Kuklová J., Přibyl O. (2019) "Framework Model in Anylogic for Smart City Ring Road Management", Smart City Symposium Prague (SCSP), Prague, Czech Republic. pp 1-5. https://doi.org/10.1109/SCSP.2019.8805681

[19] Ki Y., Na B., Kim B. (2019) "Travel Time Prediction-Based Routing Algorithms for Automated Highway Systems". IEEE Access. Vol 7. pp 121709-121718. https://doi.org/10.1109/ACCESS.2019.2937826

[20] Shi R., Zheng S., Zhang C., Zhang L. (2014) "Study on EV charging station location planning based on the load balance principle with agent-based AnyLogic simulation", 26th Chinese Control and Decision Conference (2014 CCDC), Changsha. pp 1515-1519. https://doi.org/10.1109/CCDC.2014.6852407

[21] Lupin S., Shein T., Chakirov R., Vagapov Y. (2015) "Work-in-progress:Headway optimisation in competitive public transport systems", 1st International Conference on Industrial Networks and Intelligent Systems (INISCom), Tokyo. pp 172-176. https://doi.org/10.4108/icst.iniscom.2015.258405

[22] Grigoryev I. (2015) AnyLogic 7 in Three Days: A Quick Course in Simulation Modeling, 2nd Edition. CreateSpace Independent Publishing Platform, ISBN 978-1508933748

[23] Borshchev A., Grigoryev I. (2020) The Big Book of Simulation Modeling. Multimethod Modeling with AnyLogic 8. AnyLogic North America series

[24] Боев В.Д. (2016) Моделирование в AnyLogic. Пособие для практических занятий. Военная Академия Связи

[25] Benčat G. (2020) Simulácia dopravného procesu v univerzálnom modelovacom nástroji AnyLogic. Bc thesis, No. 28260220201002, DCIS FEIT UNIZA Žilina. 40 pages

[26] Using AnyLogic Help System. https://help.anylogic.com/index.jsp (access date: 16/06/2020)