MULTIAGENT MODEL OF RAIL-ROAD INTERSECTION WITH CONNECTED VEHICLES

Aleš JANOTA[1], **Vojtech ŠIMÁK**[2], **Jozef HRBČEK**[3]

[1] University of Žilina, Faculty of Electrical Engineering and Information Technology, Univerzitná 1, 010 26 Žilina, Slovakia, ales.janota@uniza.sk
[2] University of Žilina, Faculty of Electrical Engineering and Information Technology, Univerzitná 1, 010 26 Žilina, Slovakia, vojtech.simak@uniza.sk
[3] University of Žilina, Faculty of Electrical Engineering and Information Technology, Univerzitná 1, 010 26 Žilina, Slovakia, jozef.hrbcek@uniza.sk

Streszczenie – The multiagent approach to modelling, traditionally dedicated for distributed systems, can be applied on any platform where there are more processes or control threads. The world of surface transport is a typical example of such a situation where high numbers of dynamic entities (agents) interacting with each other represent a complex problem to solve, analyse and visualise. The main focus of this paper is on functional description of the traffic control problem at the rail-road intersection. Unlike conventional approaches, this model assumes usage of modern (infrastructure-to-vehicle, vehicle-to-vehicle) communication technologies as an essential base of cooperative intelligent transportation systems. The authors use the development toolkit NetLogo, explaining step-by-step the key programming details, to get a comprehensive overview of the operation of the entire system through simple definitions of a number of simple cooperating agents. The introduced model is implementation free and shows newly offered functionalities on the principal level, while a minimum theory of collective intelligence hidden in the background is needed.

Keywords – control, model, multiagent, NetLogo, traffic

INTRODUCTION

Transportation plays an important role in society and makes a key contribution to the economy. Its future is primarily focused on new functionalities being sought with support of newly developed control, information and communication technologies. European road safety has improved greatly in recent decades. Nevertheless, the number of deaths and injuries is still far too high and progress has slowed. For the next decade, the EU has set in the EU road safety policy framework 2021-2030 a new 50% reduction target for deaths and has set out ambitious road safety plans to reach zero road deaths by 2050 ('Vision Zero'). If automated technology reduces or eliminates human errors, as is generally expected, then benefits for road safety may be substantial. We are at the beginning of the transport revolution with radical new mobility technologies that will create future road transport, significantly different from the current one. All these technologies are complementary and will merge completely over time, opening new opportunities and research challenges for building and operation of future road infrastructures. The attention of the authors of this paper is focused on railway level crossings (also railroad grade crossings) that constitute a significant safety concern internationally.

PROBLEM DEFINITION[1]

Taking advantage of the opportunities from automation technology and other emerging transport technologies is essential in order to address persistent challenges and implement additional functionalities to the conventional traffic control systems (in this case – level crossing installations) in order to increase the quality of the transport process, especially its safety aspect. The aim is to present a model of functional behaviour of the traffic participants (vehicles, train(s)) at the rail-road intersection, extended for some of these emerging technologies. The model development is based on a multiagent approach. The authors provide essential details on programming the model with support of the development environment NetLogo. It

makes possible to get a complex view of the entire system operation through simple definitions of partial cooperating agents.

The remainder of the paper is organized as follows: Section I summarizes the state-of-art and necessary background for the indicated problem solution. Section II deals with the own model development and explanation of its essential steps. Section III summarizes achieved results and discussion. The conclusions are summarized in the final Section IV.

I. BACKGROUND

A Level Crossing (LX) remains as one of the highest risk assets within the railway system often depending on the unpredictable behaviour of road and footpath users [1]. In 2017, there were 108,385 level crossings (active and passive) in the 28 EU Member States (European Union Agency for Railways, 2020). Collisions between road and rail vehicles have severe consequences such as high levels of injuries and fatalities, and significant financial losses [2]; therefore, they are a primary concern for railway authorities and the public at large. Since researchers recognize this, slowly growing attention has been given to the development of the next generation of hybrid safety systems dedicated to LXs [2]. They are based on Cooperative Intelligent Transportation Systems (C-ITS), that facilitate cooperative, connected and automated mobility [3]. This domain has the potential to form seamless Vehicular Ad-hoc Networks (VANETs) on roads and rail networks together, to facilitate safety-based Inter-Vehicle Communications (IVC) and Inter-Vehicle Positioning (IVP) among cars and trains. A high level illustration of the interface between the rail and road systems is illustrated in Figure 1.
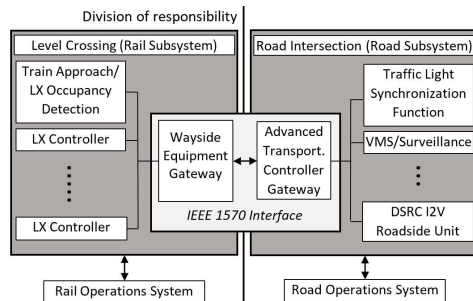

Fig. 1. IEEE 1570 based approach [4]

The paper [5] identifies the key obstacles impeding the adoption of ITS systems for the different implementation scenarios and a possible path forward towards the adoption of ITS technology. The report [6] identifies emerging C-ITS applications, including the problem of railway LX warning within the domain 'Collision avoidance and hazard detection'. Connected Vehicles (CV) are vehicles that use wireless technology to communicate with their internal and external environments. Based on involved entities (other vehicles, road side units, pedestrians, etc.) we can distinguish certain scenarios of vehicular communication such as

vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), vehicle-to-cloud (V2C), or even everything-to-everything (X2X). The study [7] aims to review and analyse literature related to data exchange in V2V communications systems. The paper [8] deals with concepts, technologies and architectures in the Internet of Vehicles. Transformation of cooperative communication technology to vehicular networking has brought a new paradigm known as a Cooperative Vehicular Networking (CVN) which has emerged as a result of advancements in wireless technologies, ad-hoc networking, and the automobile industry. A complex survey and classification of existing state-of-the-art literature on CVN can be found in [9]. The papers [10-11] generally review vehicular communication systems. The same topic reviewed from the security point of view is available in [12]. Specific considerations on how to increase the safety of LXs could be improved using modern technologies are available in [13-15] and very comprehensively discussed in deliverables of several recent projects: SNCF projects [16], Convex [17], X2Shift1 [18], SAFER-LC [19] and MOVINGRAIL [20].

II. MODEL DEVELOPMENT

Multiagent approach is one of popular modelling approaches applied in complex systems modelling [21-23]. In our context the multiagent approach is understood as a supplemental approach to modelling complex systems. An *agent* is an autonomous computational individual or object with particular properties and actions. *Agent-based* modelling is a form of computational modelling whereby a phenomenon is modeled in terms of agents and their interactions [24]. There are various agent-based modelling toolkits available, either free open-source or commercial ones, e.g. Swarm, Repast, MASON, AnyLogic, Ascape, Breve, Cormas, MASS, PS-I, SeSam, NetLogo and others. Our model is based on NetLogo, an open-source toolkit that is freely available at http://ccl.northwestern.edu/netlogo/, together with many built-in models and community models, and supported by the textbook [24]. The latest version of the toolkit (chosen for this project) is NetLogo 6.2.0 [25].

METHODOLOGY

Methodology is briefly summarized in Figure 2. The content of the Phase 1 was the study of emerging technologies, their potential and already known cases (relevant research programs, pilot projects, the current state of the literature). The next Phase 2 was focused on study of existing model library, re-usable models or sub-models and their adaptation to the starting point - a basic model into which advanced functions will be implemented. The Phase is the core phase for model development, bringing the main contribution − a novel rail-road intersection model. In the last Phase 4, the model was tested, studied and all pros and cons could be identified and summarized.
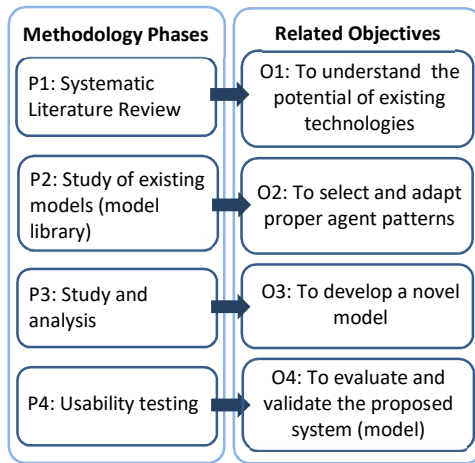
Methodology Phases — Related Objectives

Methodology Phases	Related Objectives
P1: Systematic Literature Review	O1: To understand the potential of existing technologies
P2: Study of existing models (model library)	O2: To select and adapt proper agent patterns
P3: Study and analysis	O3: To develop a novel model
P4: Usability testing	O4: To evaluate and validate the proposed system (model)

Fig. 2. 'Methodology-Research Objectives' relations

MODEL DEVELOPMENT

As a starting point (a default model) selected for further development we have chosen the model [26], coming from Chapter Five of the book [24]. This model models the movement of cars on a highway. Each car follows a simple set of rules: it slows down (decelerates) if it sees a car close ahead, and speeds up (accelerates) if it doesn't see a car ahead. The model, as well as any updates to the model, can also be found on the textbook website: http://www.intro-to-abm.com. Since the original model is well documented, the original procedures will not be discussed here and we will leave their possible explanation to the reader within his/her self-study.
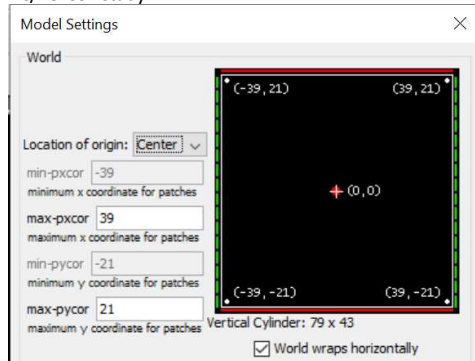

Fig. 3. Setting of the Virtual World

For the purpose of our research, some parts of the original model were intentionally removed to keep model as simple as possible. Particularly, the procedures **'to adapt'** and **'to adaptive-go'** were deleted (together with the adaptive button **'adaptive-go'** placed in *Interface tab*); and the command **'watch sample-car'** was temporarily disabled by changing it to a comment text. As a next step in the Phase

2 we changed the size of the virtual world to the matrix sized 79 x 43 patches (Figure 3) to create the sufficiently large modelling space. The model settings also involved setting of the following parameters: patch size 10, font size 10 and frame rate 30. The one-way lane has been extended for another lane to make running of two opposite traffic flows possible. The layout was re-configured using the NetLogo code in *Code tab*, as shown in Figure 4.

```
to setup-road ;; patch procedure
   set pcolor 63 ; drawing a green background
      if pycor > -1 and pycor < 2
         [set pcolor 0] ; drawing 2 black lanes
      if pxcor = 0
      [set pcolor 82] ; drawing rail line
end
```

Fig. 4. Procedure 'setup-road'

The original procedure **'setup-cars'** has also been doubled and due to establishing of two opposite flows both procedures have been differentiated by the names **'setup-cars-w'** (for car agents moving from left to right we have assigned a virtual west-east direction) and **'setup-cars-e'** (movement in a virtual east-west direction). For the sake of illustration, Figure 5 summarizes a new procedure for east-west (legt-right) direction. As a default-shape of the car in this new direction, we created a new turtle shape (here denoted **'carr'**) by horizontal flipping of the original built-in **'car'** shape, using the Turtle Shapes Editor. The dual procedure **'setup-cars-w'** looks analogically.

```
to setup-cars-e
   if number-of-cars > world-width
   [   user-message (word
        "There are too many cars for the amount of road"
        "Please decrease the NUMBER-OF-CARS slider to below"
        (world-width + 1) " and press the SETUP button again"
        "The setup has stopped.")
      stop
   ]
   set-default-shape cars "carr"
   create-cars number-of-cars
   [
      set color yellow
      set xcor random-xcor
      set ycor 1
      set heading 270
      ;; set initial speed to be in range 0.1 to 1.0
      set speed 0.1 + random-float 0.9
      set speed-min 0
      separate-cars
      set acceleration init-acceleration
      set speeds-in-test []
   ]
end
```

Fig. 5. Procedure 'setup-cars-e'

The last modification of the original model consisted in commenting (and thus disabling) the operation of watching a sample car (Figure 6). If desired, removing the comment mark ";" makes the command again fully functional.

The last necessary change was establishing a new **'breed'** (Figure 7). This keyword can only be used at the beginning of the *Code tab*, before any procedure definition. It defines a so-called **breed**, having two inputs. The first input defines the name of the agent set associated with the breed. The second input defines the name of a single member of the breed.

```
to setup
.....

;  watch sample-car  ; for watching a
                     ; red car remove 1st ";"
.....
end
```

Fig. 6. Disabling the command 'watch sample-car'

```
breed [ cars car]
```

Fig. 7. Establishing a new breed

In the Phase 3 was focused on developing our own LX model, performed a series of the following steps:

A. Configuration of the rail line: the railway line was led from the virtual south to north (from bottom to top). For the sake of simplification we assumed movement of the train in one direction only along to the single-track line. The code for drawing the railway track is given in Figure 8. In addition, the name of the procedure **'setup-rail'** must also be referenced inside the main **'setup'** procedure.

```
to setup-rail ;; patch procedure
    if pxcor = 0
        [set pcolor 82] ; drawing rail line
    if (pxcor >= -1 and pxcor <= 1)
        and (pycor >= -1 and pycor <= 2)
        [set pcolor 82] ; drawing LX area
end
```

Fig. 8. Configuration of the railway track

The results can be seen in Figure 9. For better orientation in the model, coordinates **[pxcor, pycor]** of key patches important for future settings have been manually added and highlighted (in Figure, not in the model).
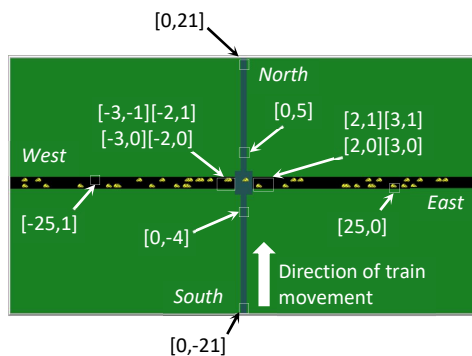

Fig. 9. The virtual world configuration

B. Activation of the warning mode: The rail line has been set established between the patches **[0, -21]** and **[0, 21]**, virtually representing south and north. The patch **[0, -4]** has been chosen to simulate a wheel detector responsible for detection of train presence (track occupancy).

Alternatively, we could use a group of selected patches (for example, **[0, -4][0, -3][0, -2][0, -1]**) to simulate an approach section in real life corresponding to a track circuit. After changing the state of this particular patch (alternatively a group of patches), the control system of the LX installation gets to the warning mode.

C. Deactivation of the warning mode: the fact that the train has already crossed the level crossing is simulated by occupancy of the patch **[0, 5]**, corresponding to the virtual wheel detector. The introduction of a one-way line has simplified the situation. In real life on a two-way track, the control algorithm would enter the so-called annulation state, when the function of possible triggering of the LX warning mode is suppressed (actively used for situations when a train approaches from the opposite direction).

In our model, the global binary variable p represents the fact whether the LX area is vacant ($p = 0$) or occupied ($p = 1$), see Figure 10. In addition, the name of the procedure **'train-presence'** must also be referenced inside the main **'setup'** procedure.

```
to go
...
    ask trains [fd 0.2 ]
    ask trains [if ycor >= 21 [die]]
...
end

to train-presence
; a train is approaching and occupying LX area
    ask trains at-points [[0 -4]]  [set p 1]
...
; a train passed the LX area
    ask trains at-points [ [0 5] ] [set p 0]
end
```

Fig. 10. Detection of train presence

D. Train generation: The train agent will enter the world at the patch **[0, -21]**, will continue to the top (north) of the world and **'die'** at the patch **[0, 21]** (see Figure 10).
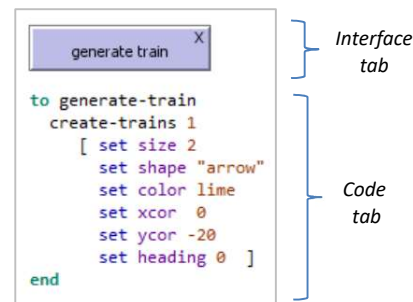

Fig. 11. Model settings used to generate a new train

The decision to generate a new train is up to the user. The train runs at constant speed defined by the command **'ask trains [fd 0.2]'** (or **'ask trains [forward 0.2]'**). For the purpose of our model, a control button has been set up in the *Interface tab* and the corresponding procedure in the

Code tab (Figure 11). Practical usage of the **'train'** agent in designed procedures has also required establishing a new **'breed [trains train]'**, similarly as we made it if for cars in Figure 7. There is no limit in how many trains a user is able to generate within the same virtual world.

E. Triggering the warning signals: To visualize activation of the warning state of the LX system to cars (and a user as well), the virtual warning signals have been generated using the patches with coordinates **[-3, -1]** and **[3, 2]**. Their positions are not highlighted in Figure 9, however, a reader can find them in later figures. The corresponding code has been inserted into the block of the procedure **'train-presence'** (Figure 12). The notation of **'signal w-e'** means "the warning signal for road vehicles running from west to north"; analogically for **'signal e-w'**. Alternatively, if needed, the signals could be displayed permanently, with variable colour representation, e.g. stable or flashing red for warning and some other colour for the idle states.

```
to train-presence
    ...
    ask patches [if (p = 1 and pxcor = -3 and pycor = -1)
        [set pcolor red]] ; signal w-e appearance
    ask patches [if (p = 1 and pxcor = 3 and pycor = 2)
        [set pcolor red]] ; signal e-w appearance
    ...
    ask patches [if (p = 0 and pxcor = -3 and pycor = -1)
        [set pcolor 63]] ; signal w-e cancelled
    ask patches [if (p = 0 and pxcor = 3 and pycor = 2)
        [set pcolor 63]] ; signal e-w cancelled
end
```

Fig. 12. Settings of the warning signals

F. Infrastructure-to-Vehicle (I2V) communication: Now we can implement the advanced function related to I2V (or I2C) communication.

It has been established as an optional choice – a user makes decision on its activation using the control switch **'connected'** (Figure 13). The corresponding **'ifelse'** command has also been inserted to the main setup procedure. Depending on the currently chosen position of this virtual switch, we set the second global variable to the value $c = 1$ (position **On**) or $c = 0$ (position **Off**).

Then we can extend the procedure **'to train-presence'** by the composite set of commands consisting of the following items: provided that $p = 1$ (LX in the warning mode) and $c = 1$ (I2V option is **On**), we ask a group of cars (generally called **'turtles'**) who are close to the LX area and are just leaving it, to perform several commands: allowing message transmission, communicating it to neighbour cars and changing their colours.

Figure 14 shows a typical situation. Cars leaving the LX area who are holders of message about the warning state, communicate this message to any other car agent found at neighbour patches (coming from the opposite direction).

Interface tab

```
to setup
    ...
    ifelse connected [set c 1] [set c 0]
    ...
end

to train-presence
    ...
; cars crossing LX are given information
; about activated warning state
    ask turtles [if (p = 1 and
      ((c = 1 and pxcor <= -2 and
        pxcor >= -5 and pycor = 1)
      or (c = 1 and pxcor >= 2 and
        pxcor <= 5 and pycor = 0)))
      [set message? true
        communicate
        recolor ]]
    ...
end
```

(Code tab)
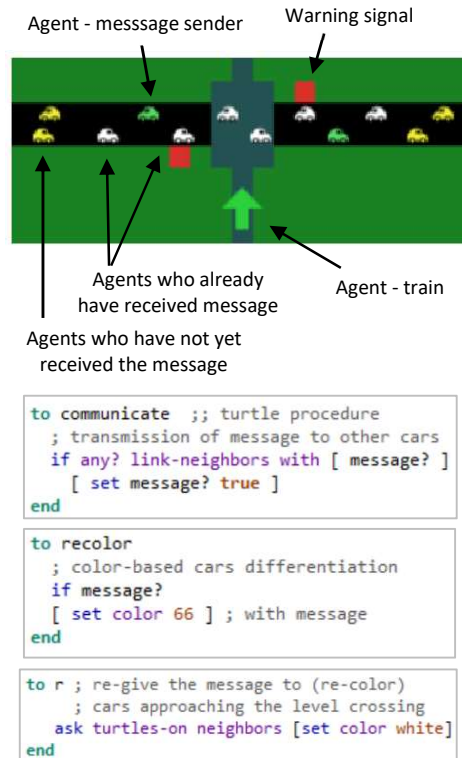
Fig. 13. Establishing I2V communication

Fig. 14. Realization and visualization of communication

```
to communicate  ;; turtle procedure
  ; transmission of message to other cars
  if any? link-neighbors with [ message? ]
    [ set message? true ]
end
```

```
to recolor
  ; color-based cars differentiation
  if message?
  [ set color 66 ] ; with message
end
```

```
to r ; re-give the message to (re-color)
      ; cars approaching the level crossing
  ask turtles-on neighbors [set color white]
end
```

G. Visualization of message communication: To visualize the communication process and to differentiate between agents-senders and agents-receivers, the following colouring convention has been accepted: agents-senders are re-coloured to green (colour code = 66) and colour of agent-receivers (who have already received the message) is set to white. Cars approaching the LX area who have not yet received the message, remain unchained, i.e. yellow.

H. Message expiration: Obviously, validity of transmitted messages must be limited either in geographic or time sense. Here, we simulate a geographical aspect of this functionality: those cars going to west (to left) who reach the patch [-25, 1] and those cars going to east (to right) who reach the patch [25, 0], will take the message invalid and stop its transmission. From the visualization viewpoint, the original colour (here yellow) will be restored at that moment. The corresponding part of the code is available in Figure 15.

[25,0]

```
to train-presence
...
; cancelling message transmission when
; reaching the selected patch
  ask turtles [if
      ((c = 1 and pxcor >= 25 and pycor = 0) or
      ( c = 1 and pxcor <= -25 and pycor = 1))
      [set color yellow
      color-sample ; re-colouring a sample car
      ]]
  ...
  end
```

Fig. 15. Geographically-based message validity

I. Cancelling I2V/V2V communication: Once the train passed the LX area and its position was detected at the patch [0, 5], the global variable *p* is given the default value 0. Then car agents who have just crossed the LX area stop obtaining messages from the LX system (**'message?'** is set to **false**), see Figure 16. The process of colouring from yellow to green also becomes stopped.

```
to train-presence
...
; cancelling I2C communication
  ask turtles [if (p = 0 and
      ((c = 1 and pxcor >= 2 and pycor = 0) or
      ( c = 1 and pxcor <= -2 and pycor = 1)))
      [set message? false
      communicate
      recolor
      color-sample ; re-colouring a sample car
      ]]
  end
```
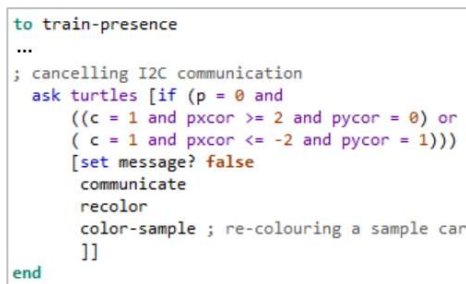
Fig. 16. Cancelling I2V communication

Note: It is necessary to mention that the original model contained a randomly generated sample-car (the only one painted in red). Since this particular agent is established to enable collecting some traffic data from the running simulation (as an optional choice set by a user), in all new procedures developed and discussed above we made our best to ensure its visibility. For example, in Figure 16 this is ensured by the procedure **'color-sample'**.

Once the model has been completed, the Phase 4 can be started. The complex view at the whole *Interface tab* is available in Figure 17. In addition to the 2D virtual world on the left side, we can see that there is a group of control elements (control buttons, sliders and switches) on the right side. Data obtained during the simulation run can be shown using various monitors, plots and outputs. These output elements visible on far right in Figure 17 are overtaken from the original model where they are explained in more details. The dependencies in the graphs are unique and result from the current settings of the control elements and the randomly generated vehicle configuration (randomly generated positions of individual cars, speeds, accelerations etc.).

III. RESULTS AND DISCUSSION

We believe that agent-based representations are easier to understand than mathematical representations of the same phenomenon. This is because agent-based models are constructed out of individual objects and simple rules for their movement of behaviour, as opposed to equational models that are constructed from mathematical symbols [25]. In this context we introduced development of a novel traffic model at rail-road intersection that takes advantages of some modern communication technologies. To keep the code modular and the model as simple and understandable as possible, we have introduced several simplifications mentioned above. However, the model can more or less easily be further extended to provide a higher level of complexity if required. To show the simulation potential, let us suggest some of them.

As far as the road traffic is concerned, to make the traffic flow more realistic, we could generate more diverse traffic flow composition (cars, trucks, motorcycles) according to defined random distribution; similarly we could work with a certain division of the generated vehicles into conventional and connected ones. However, this aspect would increase complexity of the model significantly. Another opportunity is to re-establish the original adaptation procedures available in the original overtaken model.

On the railway side, the one-way railway line could easily be changed to two-ways one with necessity to cope with a higher complexity of the control logic, e.g. answering emergency situations resulting from simultaneous occurrence of two trains running in the opposite directions – a user could have no permission to generate such a situation, or if so (intentionally or unintentionally), the control logic would have to answer properly. What else, a completely new line for the opposite direction could be added; including possibility to generate trains in both directions (manually or based on some kind of virtual time-table to be inserted). In our model the train runs at constant speed only – this could

simply be changed by adding a control slider into the *Interface tab*, setting the **'train-speed'** (for example at the scale from 0 to 1) by the user and changing the command in Figure 10 to **'ask trains [fd train-speed]'**.

Every country has specific signalling signs used at the LXs, so the simplified form used in our model could also become more realistic to make the model more user-friendly. The graphic look could also be improved by adding representation of half-gates or full-gates, if needed. More attention could also be paid to various forms of how to activate the LX warning mode (i. e. to consider more realistic settings corresponding to real track circuits, wheel (track) detectors, manual local or remote control, direct link train-LX, distances and speeds proportional to real limits, etc.).
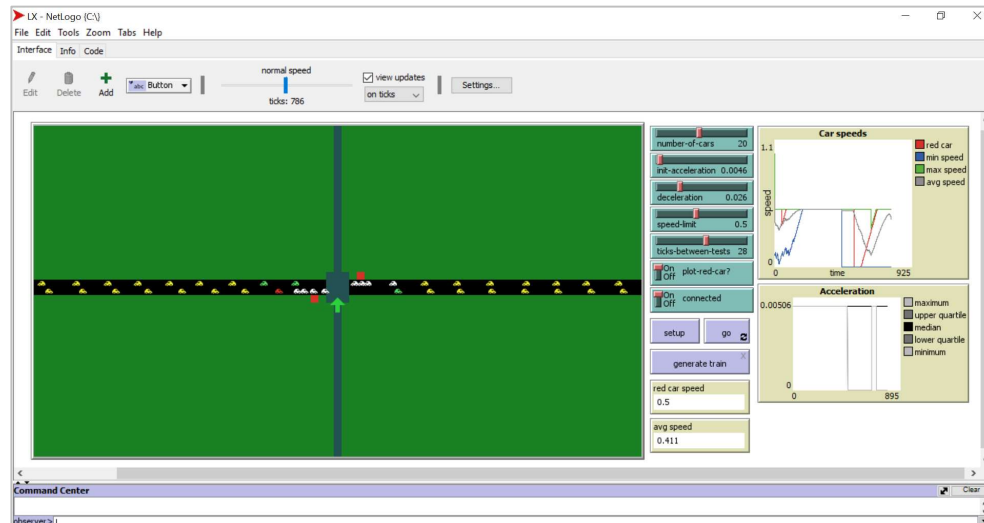

Fig. 17. Complex view at the Interface tab

Unlike our simplified "arrangement", the standard control logic would need information about train movement at least from three line sections (approaching, detection and annulation ones).

Using more advanced abilities of the toolkit, we could use **'links'** (another type of agents offered in the NetLogo environment) to directly visualize establishing communication connections between connected entities (I2V and V2V, or even directly Train2V communication) during simulated operation.

What's more, some other monitors could be established to process, evaluate and show more metric information about simulation (e.g. counting traffic intensities, traffic moment at the LX, etc.). A reader can find more inspiration in other existing models (either default models library built-in in the toolkit or models searchable in the library of community models at the Internet).

In opposition to what has been said above, however, it should be emphasized that implementation of useless details (e.g. driven by the pursuit of a more realistic look) may devaluate the model and make its practical usage more difficult.

IV. Conclusions

A multiagent system is usually said to have a potential to enhance overall system performance, specifically along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility and/or reuse. The NetLogo 6.2 toolkit has been chosen to create the model itself and the environment in which functional behaviour of the considered control system can be simulated and studied. The model developed and presented in this paper is a combination of a traditional (conventional) view at the traffic control at the road-rail intersection and a progressive view counting on usability of modern technologies. V2X communication, which involves vehicles exchanging data with each other and the infrastructure, has proven to improve traffic safety and increase the efficiency of transportation systems. Although the multiagent modularity is traditionally dedicated for distributed systems, it is advantageous to apply it on any platform where we have more processes or threads, which is the case of our problem. The modelled problem demonstrated how with a minimum theory of collective intelligence hidden in the background a complex control problem can be solved.

The model itself is implementation free and shows newly offered functionalities on the principal level only. Readers who are interested in running it, studying and making their own experiments are welcomed to install the free multiagent development toolkit [26] and download the source code, publicly available at the link http://kris.uniza.sk/janota/dokumenty/LX.nlogo.

After opening the model in the toolkit, the *Info tab* provides other useful information, summarizing answers to a set of commonly given questions (such as how to use the model, things worth of noticing, things to try, how to extend the model, utilized toolkit features, relation to other existing models, acknowledgments, copyright and citation details, etc.).

REFERENCES

[1] Fayyaz M. A. B., Alexoulis-Chrysovergis A. C., Southgate M. J., Johnson Ch. (2020) "A review of the technological developments for interlocking at level crossing". Proc IMechE Part F: J Rail and Rapid Transit 0(0), pp. 1-11, https://doi.org/10.1177/0954409720941726

[2] Ansari K. (2014) Development of an Inter-Vehicle Communications & Positioning Platform for Transport Safety Applications. PhD. Thesis, Queensland University of Technology, 222 p.

[3] Karkhanis P., Brand M. G. J. v. d. , Rajkarnikar S. (2018) Defining the C-ITS Reference Architecture. In IEEE Int. Conf. on Software Architecture Companion (ICSA-C), Seattle, WA, 2018, pp. 148-151, doi: 10.1109/ICSA-C.2018.00044

[4] IEEE Std 1570-2002 (2002) IEEE Standard for the Interface Between the Rail Subsystem and the Highway Subsystem at a Highway Rail Intersection. pp. i-65

[5] Wullems Ch., Wayth R., Galea V., Nelson-Furnell P. (2014) In-vehicle railway level crossing warning systems: can intelligent transport systems deliver? In Schweiger, R. (Ed.) Proceedings of the 2014 Conf. on Railway Excellence: Rail Transport for Vital Economy. Railway Technical Society of Australasia (RTSA), Australia, pp. 1-12.

[6] Logan D. B., Young K., Allen T., Horberry T. (2017) Safety Benefits of Cooperative ITS and Automated Driving in Australia and New Zealand. Austroads, Research Report AP-R551-17, 51 p.

[7] Ameen H. A., Mahamad A. K., Zaidan B. B., Zaidan A. A., Saon S., Nor D. M., Malik R. Q., Kareem Z. H., Garfan S., Zaidan R. A., Mohammed A. (2019) "A Deep Review and Analysis of Data Exchange in Vehicle-to-Vehicle Communications Systems: Coherent Taxonomy, Challenges, Motivations, Recommendations, Substantial Analysis and Future Directions".

[8] IEEE Access, vol. 7, pp. 158349-158378, doi: 10.1109/ACCESS.2019.2949130

[9] Mahmood Z. (2020) Connected Vehicles in the IoV: Concepts, Technologies and Architectures. In Connected Vehicles in the Internet of Things. Springer International Publishing, pp. 3–18, https://doi.org/10.1007/978-3-030-36167-9_1

[10] Ahmed E., Gharavi, H. (2018) "Cooperative Vehicular Networking: A Survey. IEEE Trans. Intell. Transp. Syst. 19, pp. 996–1014. https://doi.org/10.1109/TITS.2018.2795381

[11] Riaz F., Niazi M. A. (2016) " Road collisions avoidance using vehicular cyber-physical systems: a taxonomy and review". Complex Adapt Syst Model, vol. 4, no. 15, https://doi.org/10.1186/s40294-016-0025-8

[12] Papadimitratos P., La Fortelle A. D., Evenssen K., Brignolo R., Cosenza S. (2009) "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation". IEEE Communications Magazine, vol. 47, no. 11, pp. 84-95, November 2009, doi: 10.1109/MCOM.2009.5307471.

[13] Sheikh M. S., Liang J., Wang W. (2020) " Security and Privacy in Vehicular Ad Hoc Network and Vehicle Cloud Computing: A Survey". Wireless Communications and Mobile Computing, vol. 2020, Article ID 5129620, pp. 1-25, https://doi.org/10.1155/2020/5129620

[14] Wullems Ch., Wayth R., Galea V., Nelson-Furnell P. (2014) In-vehicle railway level crossing warning systems: can intelligent transport systems deliver? In Conf. on Railway Excellence, Adelaide, 5 – 7 May 2014, 12 p.

[15] Parikh G., Duhn M., Hourdos J. (2019) How Locals Need to Prepare for the Future of V2V/V2I Connected Vehicles. Final research report MN/RC 2019-35, 135 p.

[16] Wang X., Li J., Zhang C., Qiu T. Z. (2019) "Active Warning System for Highway-Rail Grade Crossings Using Connected Vehicle Technologies". Journal of Advanced Transportation, vol. 2019, Article ID 3219387, 11 p., https://doi.org/10.1155/2019/3219387

[17] Taillandier V. (2018) Smart level crossing. In 16th Int. Conf. on Intelligent Transportation Systems Telecommunications (ITST), Lisboa, pp. 1-5, doi: 10.1109/ITST.2018.8566824.

[18] ConVex (2020) Connected Vehicles (V2X) of Tomorrow. Project Deliverables. https://convex-project.de/deliverables.html

[19] X2Rail-1 (2019) Start-up activities for Advanced Signalling and Automation Systems. Deliverable D7.1 Analysis of existing lines and economic models, 165 p.

[20] SAFER-LC (2019) SAFER Level Crossing by integrating and optimizing road-rail infrastructure management and design. Deliverable D4.4 – Results of the evaluation of the pilot tests , 163 p.

[21] MOVINGRAIL (2020) Deliverable D3.2: Advances in Automated Vehicle Technology and Applicability to Railways. 58 p.

[22] Derakhshan F., Yousefi S. (2019) "A review on the applications of multiagent systems in wireless sensor networks". International Journal of Distributed Sensor Networks, Vol 15, No. 5, pp. 1-19, https://doi.org/10.1177/1550147719850767

[23] Sathiyaraj R., Bharathi A. (2020) "An efficient intelligent traffic light control and deviation system for traffic congestion avoidance using multi-agent system". Transport, Vol. 35, Issue 3, pp. 327–335, https://doi.org/10.3846/transport.2019.11115

[24] Etzioni A., Etzioni O. (2016) "AI assisted ethics". Ethics and Information Technology, Vol. 18, No. 2, pp. 149-156, https://doi.org/10.1007/s10676-016-9400-6

[25] Wilensky U., Rand W. (2015) An Introduction to Agent-Based Modeling. Modeling Natural, Social, and Engineered Complex Systems with NetLogo. Cambridge,MA. MIT Press, 482 p.

[26] Wilensky U. (1999) NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/

[27] Head B., Rand W., Wilensky U. (2015) NetLogo Traffic Basic Adaptive Individuals model. Center for Connected Learning and Computer-Based Modeling, Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/models/TrafficBasicAdaptiveIndividuals