AVIATION
ISSN: 1648-7788 / eISSN: 1822-4180
2022 Volume 26 Issue 2: 104–111
https://doi.org/10.3846/aviation.2022.16869

AN EMULATION ORIENTED METHOD AND TOOL FOR TEST OF GROUND TRAFFIC CONTROL SYSTEMS AT AIRPORTS

Farid SAIFUTDINOV, Ilya JACKSON, Jurijs TOLUJEVS *

Faculty of Engineering, Transport and Telecommunication Institute, Riga, Latvia

Received 19 January 2021; accepted 13 August 2021

Abstract. The paper discusses the prospects for the development and implementation of centralized ground traffic control systems at airports. The automatic control system can only work if there is accurate data on the location of mobile objects, which include both vehicles involved in the maintenance of aircraft and the aircraft themselves. In order to develop and test software for any specific centralized control system, the emulation mode should be used, in which the simulation model of the airport transport network works in conjunction with the real control software. In this case, one of the main functions of the simulation model is the generation of data streams that appropriately reflect the processes of movement of objects in the transport network of a specific airport. The paper describes a universal simulation program that allows one to simulate precisely described scenarios for the process in a transport network, which necessitates decision-making at the level of a centralized control system. The movement of objects in the model is accompanied by the recording of their coordinates in the Digital Twin. In this way, real streams of measurement data from various systems for determining the position of moving objects are modeled and stored.

Keywords: airport, ground traffic, centralized control, simulation, emulation, data flows, digital twin.

Introduction

The Vision 2050 announced in 2011 by the International Air Transport Association (IATA) predicts that airports of the nearest future will incorporate automatic and automated systems for centralized control of ground vehicles (GV) and aircraft (AC) (IATA, 2011). The centralized control system will not duplicate or cancel the up-to-date functions of onboard local control systems because it will solve traffic control problems of the transport system as a whole. Functions such as keeping distance between vehicles and avoiding collisions with physical obstacles will continue to be performed by local onboard control systems. Only in dangerous or emergencies should such functions be taken over by the centralized control system. Nowadays, centralized surface traffic control functions at airports are performed by ordinary airport controllers or specialized controllers that control only surface traffic. There are convincing reasons to assert that Automatic Centralized Surface Movement Control Systems will be more reliable and efficient than manual control systems because they will be able to perceive and interpret such amounts of information that significantly exceeds the human-dispatcher capabilities (Augustyn & Znojek, 2015).

The most important document defining the direction of development of surface movement guidance systems at airports is (EUROCONTROL, 2020), which describes the services provided by the Advanced-Surface Movement Guidance and Control System (A-SMGCS) for the Single European Sky (SES). These services include Surveillance, Airport Safety Support, Routing, and Guidance. The main objective of the Surveillance Service is to provide the dispatcher with visual information about the location of all participants in surface traffic on the territory of the airport. The Airport Safety Support Service generates warnings for dispatchers if a dangerous approach of objects is detected on the runway or on sections of the transport network along which AC or GV are moving. The Routing Service develops a specific route for each mobile unit (AC or GV) that connects the given start and end point of the trip. Finally, the Guidance Service shows the AC crew or the driver of the GV the route of movement using chains of lights on the surface of the earth.

A-SMGCS services can significantly facilitate the workflow of dispatchers, but their evolution does not necessarily imply a transition to fully automatic ground traffic management. Moreover, EUROCONTROL documents are not binding, and each airport makes its own

*Corresponding author. E-mail: *tolujevs.j@tsi.lv*

decision regarding the timing and scope of the A-SMGCS application. It can be argued that the creation of automatic ground traffic control systems is one of the development goals for systems similar to A-SMGCS, and this goal will certainly be achieved someday, but due to the incredible complexity of the tasks arising on this path, no one is attempting to set a deadline to which these tasks can be solved.

These tasks can be conditionally divided into the areas of hardware and software, similar to how it is done in the domain of computer science. The main task in the field of hardware is to create "ideal" systems for determining the location of moving objects. The term "ideal" refers to the systems that a) in all weather conditions, b) for any number of controlled AC and GV, c) with an error not exceeding 1 meter, d) with sufficient frequency to control objects is capable of transferring the coordinates (x, y, z) of all moving objects to the computer of the centralized control system. There has been continuous progress in the field (Saifutdinov & Tolujevs, 2020a), and there is no doubt that "ideal" object positioning systems will be created one day. It is quite clear that only systems in which different methods and technologies of position determination are applied simultaneously will be able to fulfill all the requirements listed above. For airport services, the range of facilities extends from conventional surface movement radar (SMR) systems to Intelligent Cameras-based systems (Dimitropoulos et al., 2005) and associated Video Analytics techniques. New types of satellite and inertial navigation systems, as well as LiDAR devices, will be installed on board mobile objects (Lee et al., 2020). An important aspect of hardware development is also progressing in the field of the Unmanned Ground Vehicle (UGV) that in the foreseeable future will replace the Human Driven Vehicle (HDV), which performs the functions of GV at airports.

This work is related to the tasks that need to be solved within the domain of software. The main functions of the new generation of software become clear if, in the above description of the functions of the A-SMGCS, the human dispatcher is replaced by a dispatch computer program. Such a program can be considered "ideal" if it is capable of a) constant monitoring of the location and status of all participants in surface traffic, b) prediction of the possibility of the emergence of critical situations (dangerous approach of objects, etc.), c) identification of already existing critical situations, d) identification of traffic participants that may be affected by a critical situation and sends them specific control commands or re-routing data.

This paper does not concern the development of control programs themselves, but it proposes a tool for testing them, focused on the use of simulation models in emulation mode (Xcelgo, 2020). Since there is practically no information on the application of this method in the field of aviation, the first part of the work considers examples of testing programs for centralized control of transport processes in areas where there is already useful experience in solving such problems. The first group of examples relates to the field of Automated Guided Vehicles (AGVs) and the second to the use of the Floating Car Data (FCD) concept for road traffic control. Taking into account the experience of specialists from these areas, a rationale is given for the choice of a method for software implementation of its own product. In the second part of the work, a program developed by the authors for simulation modeling of transport processes is described, which is entirely focused on joint use with centralized ground movement control programs at airports.

1. Application of simulation models in emulation mode for debugging control software

The concept of emulation is well known to specialists involved in developing and testing software designed for automatic control of technical systems. The essence of emulation is that the control object is replaced by a simulation model that interacts with real control programs (Xcelgo, 2020). The model includes a sensor part, from where signals are sent from various sensors corresponding to the devices of the actual system. Control programs process these signals and generate control commands that are perceived by the executive part of the model. The model responds adequately to control commands; that is, the same changes and events occur in it as should occur in a real system. Usually, the simulation model is processed on a separate computer that must be hardwired to the computer on which the control programs are installed.

Emulation can be used to test new control programs and adapt standard programs before connecting them to a new object. The main effect of the use of emulation is that the time for debugging and commissioning complex control programs is significantly reduced. Naturally, this effect is achieved provided that the processes in the simulation model proceed much faster than in the real system. A large economic effect is achieved, for example, in the case of using emulation when commissioning a management system for a large automated warehouse (Spieckermann et al., 2012). Without emulation, debugging programs can take several months, during which high costs arise due to the need to keep the warehouse equipment in working order constantly.

The work (Helleboogh et al., 2006) discusses the issues of testing software for controlling a group of Automated Guided Vehicles (AGV) in a large warehouse. Each AGV has onboard software that, for example, prevents the AGV from colliding with each other or with other obstacles. However, in the transport network, conflict situations can arise that can only be resolved by activating centralized control programs. The system has a central server that schedules the AGV movement, sends AGV commands, and constantly polls their status. The authors of the work report on using a warehouse simulation model for testing both onboard and central software, which together provide reliable AGV motion control.

A special kind of AGV is Automated Rail-Mounted Gantry Cranes (ARMG), which are often used in container terminals in seaports. Such cranes serve container storage areas and operate in automatic mode. In large ports, as a rule, several dozen ARMG-type cranes are used. Rintanen and Thomas (2021) report on the experience of using simulation and emulation when debugging ARMG crane control software in ports such as CTA and CTB in Hamburg, APM Terminal Virginia, Antwerp Gateway, London Gateway, Khalifa Container Terminal, and others. The simulation model was used as a substitute for a real crane. Emulation allows debugging the control software simultaneously with the production and installation of equipment and creates a new environment for controlling the quality of the software product.

Floating Car Data (FCD) is a method of collecting data on the current state of road traffic. The data comes from the so-called "connected vehicles" in the form of records, the main content of which is the time stamp and coordinates of the location of the vehicle. In Astarita et al. (2020), not only the experience of the practical application of a traffic management system based on the FCD method is reported, but also the process of developing and debugging the corresponding software, in which simulation and emulation were used, is described in detail. At all stages of this project, experiments were carried out on the basis of the TRITONE software platform, specially developed for the purposes of simulation and emulation. The choice of the software implementation method of the TRITONE system seems interesting, which will be discussed in more detail in the following section.

2. Justifying the choice of the process's simulation method at airports

Making a choice among various tools for simulating transport processes at the airport, software products belonging to four classes were taken into consideration:

1) airport simulation software (e.g., TAAM, ArcPORT, SIMMOD and CAST);
2) road traffic simulation software (e.g., PTV Vissim or SUMO);
3) general-purpose simulation software (e.g., AnyLogic or Simio);
4) general-purpose programming languages (e.g., C #, C ++, Java, JavaScript, Python and Visual Basic).

Since the products of the last three classes are widespread and their properties are well known to many specialists, the features of the most famous and contemporary products in the field of airport simulation software are emphasized. All of them are focused on modeling processes in such areas of the airport as airspace, airside, landside, apron and terminal, but the following overview will highlight the properties of these packages associated with ground transport processes.

For example, the Total Airspace and Airport Modeler (TAAM) package defines the following position types for AC locations: Gates, Device Stations, Long Term Parking Positions, and Standoffs (Jeppesen, n.d.). In order to simulate AC movement, such path types as Taxiways, Runways, and Pushback Paths are used. The capabilities to model GV are closely associated with aprons. The package developers call it a virtual laboratory, in which any operational environment can be simulated, and the results of simulation experiments can be analyzed.

The ArcPORT Airside module provides a platform for assessing processes at the aerodrome field as well as in its airspace (ArcPORT, n.d.). In particular, it is applied to study the effect of incoming and outgoing AC routes on taxiing time, control algorithms for taxiways and runways, and to estimate the airport's AC input and output capacity.

Simmod PRO package! (Atac, n.d.) allows effective simulation in interactive mode, when at certain times the model stops and waits for instructions from the user, which are the result of his decisions. Such instructions are used to control the processing steps of each individual AC, taking into account the specific conditions that develop in both the airspace and the aerodrome (Bertino et al., 2011).

It should be pointed out that the TAAM, ArcPORT, and SIMMOD packages discussed above are focused on the AC motion processes, while the GV motion processes are modeled almost automatically. There is a special CAST Vehicle Ground Handling module only in the CAST package, which makes it possible to set the GV movement control strategies. The choice of specific strategies is made by specifying special input parameters of the model (Airport Research Center, n.d.).

All the packages mentioned above allow displaying AC and GV traffic processes on the airport zone with a high degree of adequacy, including photorealistic 3D animation. These packages are developed in order to solve both tactical and strategic tasks. Tactical tasks include the training of air and ground traffic controllers. Strategic objectives include comparing alternative management strategies and even options for airport transport network plans by analyzing the most important process quality indicators, which are assessed using simulation.

The modeling problem considered in this paper belongs to the group of tactical tasks. Therefore, it is associated with the study of control processes by participants in ground movement in real-time operational mode. It is assumed that a surface movement dispatcher in the future can be replaced by a computer program that works with partial human participation or entirely automatically. Dispatchers make decisions mainly based on visual information, which he or she receives directly by observing the airfield from the control tower or on the screens of monitors that receive images from TV cameras (Kaidi, 2019) or regular SMR. The automatic dispatcher can work only using data streams on the location of mobile objects. In order to develop, train and validate such an automatic dispatcher, it is necessary to generate data flows that do not reflect abstract or random situations but particular situations that require the intervention of a centralized control system. The above simulation packages are poorly suited for modeling and recording such data flows since they are expensive commercial products with closed source code, and such a mode of applying models is not provided in them even at the conceptual level.

The problem of choosing software for developing models operating in emulation mode was solved by the authors (Astarita et al., 2012), researchers at the University of Calabria. As a result of comparing the last three product classes noted above, it was decided to develop an independent TRITONE package using general-purpose programming languages. Although the competing open-source SUMO package has been available since 2001, solving research problems in traffic management at the micro level requires the development of a tool that provides even more opportunities for creating scripts that determine the behavior of each individual vehicle. In addition, the option was taken into account when the simulation and control programs must be executed on separate computers.

Since the modeling problem considered in this work involves the use of large amounts of data both at the input and at the output of the model, a decision was made that implements not only the concept of open source but also the concept of open data. A consequence of this decision was the development of a Ground Traffic Scenario Simulation (GTSS) program (Saifutdinov & Tolujevs, 2020b). The program is developed using the VBA programming language, and it supports development and processing of models in the MS Excel environment. The main advantage of such a solution is the ideal availability of all types of data: initial data, current state parameters of all model elements and simulation results. The user's interaction with the software is mainly based on the analysis of visual information, which is provided by the animation tools within the GTSS program (Figure 1).

3. Modeling scenarios with GTSS

Figure 1 illustrates the two main modes of applying the simulation model. In the "hand control mode" the user of the model manually describes the scenario of movement of a specific AC or GV. As a result of the implementation of this scenario, a planned critical situation arises in the model, which requires the intervention of centralized management. In the absence of a well-tried automatic traffic control program in the Applications block, the user of the model takes the role of the surface movement controller. He or she develops and introduces into the model a new scenario that describes the behavior of road users after the moment of making control decisions. In the "automatic control mode", the required initial scenario is also specified by the model user, but the scenario of the behavior of road users after a critical situation occurs is determined by the tested control program. This particular mode of application of the simulation model is emulation.

A stochastic approach to systems research is most commonly used in simulations (Law & Kelton, 2000). Models generate input flows of mobile objects using the specified probability distribution for the time intervals. The main numerical results of modeling are obtained by statistical processing of a large number of events and states and are presented in the form of interval estimates of mean values or in the form of histograms. Even if deterministic schedules are used to simulate the input flows, for example, the schedule of flights arriving at an airport, the model remains stochastic since many processes of servicing AC and passengers are modeled using random variables with given probability distributions. In this kind of modeling, the modeler sometimes uses the concept of a scenario to refer to the planned versions of the model inputs.

The more specific concept of a scenario is often used in traffic flow simulation using software packages such as PTV Vissim or SUMO. In this case, a scenario is usually understood as the entire set of data that describes the simulated traffic, including network data, additional traffic infrastructure (e.g., traffic lights), and traffic demand (Lopez et al., 2018). The traffic demand in vehicle flow models usually involves the stochastic component, but sometimes deterministic models are also used, for example, to assign traffic lanes for individual vehicles (Chen et al., 2020).

GTSS considers scenarios as accurate descriptions of the routes and maximal speeds of all vehicles, including AC and GV, within a certain period of time. Usually, this time does not exceed 10–15 minutes, as it is sufficient to form a critical situation in which the intervention of the

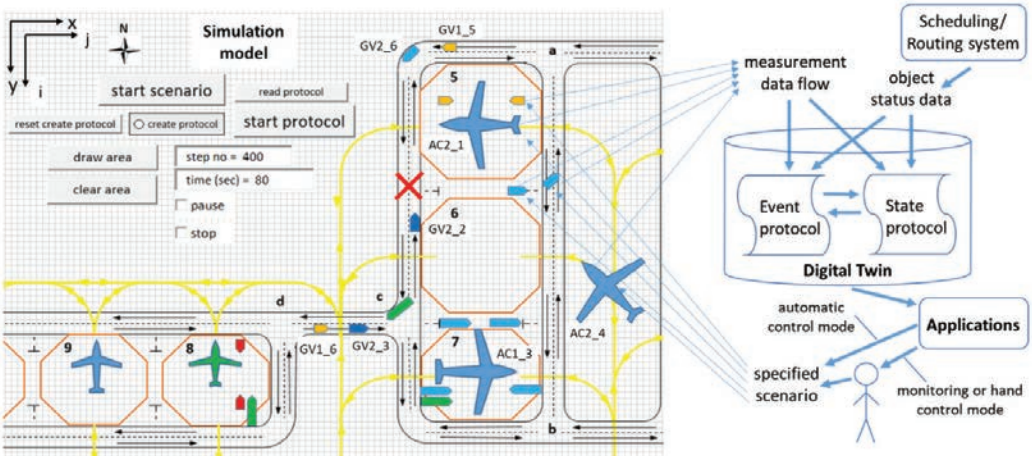

Figure 1. Fragment of the model built using GTSS

central control system is necessary. In a similar form, one can describe a scenario that shows a way to get out of a critical situation as a result of a decision made by the model user or the tested control program included to Applications (Figure 1).

The vehicle routes are described based on the division of the transport network into sections. Figure 2 shows two fragments of the transport network, which can also be seen in the airside plan view in Figure 1. 1–2 Figures show parking stands for ACs numbered 5, 6, and 7, as well as three intersections in the network for GV traffic, indicated by letters *a*, *b* and *c*. The blue arrows show the paths of the GV, and the thicker green arrows the paths of the AC. When developing a model for a specific airport, the sections of the transport network are set manually by marking the cells of the grid space on an MS Excel worksheet, which depicts the airside zone of the airport at the appropriate scale. The model retains the ability to set or read each object's coordinates (x, y) in meters in free 2D space.

In order to set the routing parameters, the user fills in the tables "Static positions" and "Dynamic routes" (Figure 3). In the "Static positions" table, the user describes stationary objects that should appear at certain points in the airport zone. The objects' names of type AC1_1 or GV3_1 correspond to specific AC or GV, which are predefined in the form of graphical Excel objects of type Shape in the model created using GTSS. In the GTSS program, one can define any number of classes of moving objects and any number of objects of each class. The "positioning time" column specifies the moment in time when the object should appear in the model. The "row number" and "column number" columns define the position of the object's reference point in the center of the corresponding cell (i, j) on the Excel sheet, and the "direction ID" column sets one of eight options for the object's orientation along the horizon (N, NE, E, SE, S, SW, W, and NW). The position of the object can also be specified using coordinates (x, y). If the user has not specified these coordinates, then they are calculated automatically when the program is called. An object added to the "Static positions" table can remain in its initial position for as long as desired. The user can start modeling the motion of this object in free 2D space by specifying new coordinates (x, y) for it in the same "Static positions" table or by incorporating it into the "Dynamic routes" table. Naturally, in the "positioning time" column, the new time value must not be less than the previous one.

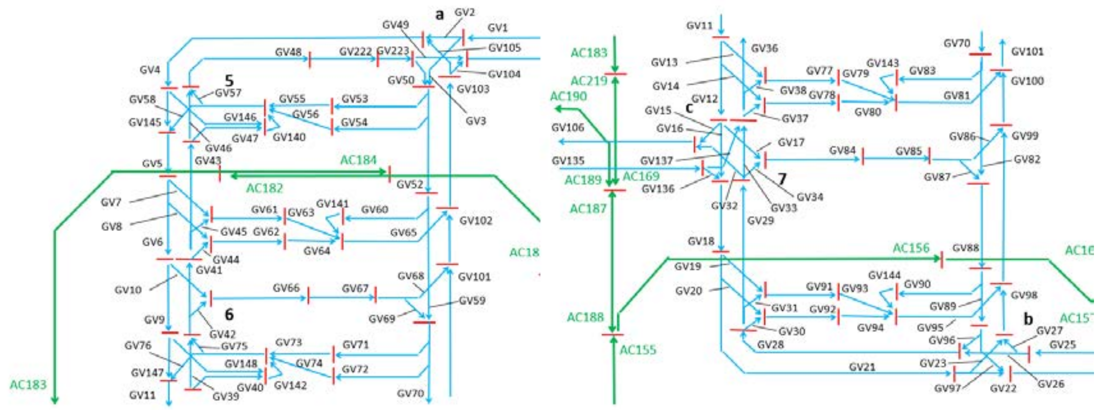

Figure 2. Topology and designation of sections of the transport network

Static positions							
positioning time	object name	object color	row number	column number	X coordinate (m)	Y coordinate (m)	direction ID
0	AC1_1	standard	31	146	695,14	310,09	W
0	AC4_1	blue	72	158	751,72	114,62	E
0	GV3_1	blue	25	157			W
0	GV2_1	standard	42	149			E

Dynamic routes															
positioning time	object name	object color	speed (m/s)	number of sections	section 1	section 2	section 3	section 4	section 5	section 6	section 7	section 8	section 9	section 10	section 11
10	AC2_1	standard	13	14	AC153	AC191	AC195	AC197	AC198	AC189	AC188	AC170	AC171	AC191	AC193
					10	8	6	4&240							
15	GV2_2	green	6	22	GV12	GV15	GV106	GV108	GV117	GV119	GV120	GV121	GV133	GV134	GV135
					5	4	5	5	5	5	4&26	3	5		5
17	GV2_3	light blue	6	24	GV26	GV28	GV29	GV32	GV106	GV108	GV117	GV119	GV120	GV121	GV133
						4	5	4	5	5	5	4&27			
21	GV1_1	light blue	6	22	GV147	GV11	GV12	GV15	GV106	GV107	GV110	GV138	GV116	GV134	GV135
							5	4	5	5	4&72				

Figure 3. Screenshot of the "Static positions" and "Dynamic routes" tables

In the "Dynamic routes" table, the user describes the trajectory of each object as a sequence of sections of the transport network. The column "number of sections" indicates the number of sections that the object must pass within the scenario. This number of sections in the model is not limited. The record type $\frac{AC153}{10}$ means that the maximum speed of an object on section AC153 is 10 m/s. The record type $\frac{GV120}{4\,\&\,26}$ means that the maximum object speed on the GV120 section is 4 m/s and at the last point of the section the object should be delayed for 26 seconds. If the speed of the object is not specified in the cell of the section, then its maximum speed is determined by the value in the column "speed (m/s)". The maximum speed is always referred to, since the actual speed can be reduced due to the inability to overtake the object in front.

4. Recording and transforming event and state protocols

Modeling and recording data flows on the location of moving objects is a technical task of GTSS, while the intellectual task is to simulate critical situations in which a dispatcher generates and checks his operational decisions. The experience of making such decisions in the future should be used to train the automatic centralized control system of transport processes at the airport.

The main steps performed by a dispatcher in modeling specific decision-making situations are as follows:

1. Dispatcher selects one of the typical critical situations that may arise in an airport ground transport system. Such a situation can be attributed to the class of explicit or recognizable situations. The situation is explicit, if the message about it comes to the dispatcher from some of the airport services. An example of such a situation is the appearance of an obstacle on one of the sections of the transport network or the appearance of a priority vehicle, which must reach the destination of the route as soon as possible. Recognizable situations include situations that the dispatcher determines for himself or herself by observing the processes in the ground transport system. An example of a recognizable situation can be a congestion of vehicles as a result of a traffic jam, congestion, dangerous proximity of objects to each other or to other physical obstacles, or dangerous excess of the speed limit.
2. Dispatcher designs a scenario for the occurrence of the critical situation, describing the behavior of all its participants using the "Static positions" and "Dynamic routes" tables. Typically, this step is implemented with several iterations until the dispatcher is satisfied with the animation being observed. The dispatcher fixes the moment in time, which he or she will consider the decision-making moment.
3. Dispatcher designs a scenario of the behavior of all in critical situation participants after the decision-making moment. In order to accomplish this, in the "Static positions" and "Dynamic routes" tables, he or she changes or adds rows, observing the condition that the time in the "positioning time" column necessarily follows the moment of making a decision or coincides with it. This also leads to the desired state through several iterations.
4. Dispatcher activates the "create protocol" option on the Excel worksheet with the model image (Figure 1) and with the "start scenario" key he or she starts the simulation process in which both scenarios described above are reproduced. The program will save a data flow regarding the location of objects in the form of the State Protocol in a text file, the name of which must be specified by the user.

Figure 4 shows the State Protocol format and a fragment of a real file containing data about two objects of type GV and two objects of type AC for time points 10.2 and 10.4 respectively. The name of the transport network section or the coordinates of an Excel cell for objects that are not in motion are shown as "location". Time in the State Protocol is discrete, since the model uses the "delta T" time counting method with a numerical step value of 0.2 seconds. As will be shown below, as a result of recording a real data stream with coordinates (x, y), a so-called Event Protocol is produced, in which messages from individual objects arrive at arbitrary times. By extrapolating the measurement data, the Event Protocol will need to be transformed into the State Protocol format with discrete time, since automatic algorithms for analyzing the current situation can simultaneously take into account the states of all objects only at certain points in time.

In a real system, the flow of primary data on the location of moving objects can come in both synchronous and asynchronous modes. Synchronous mode is observed if information comes from the SMR, since the coordinates of the objects are tied to a specific point in time corresponding to the last cycle of the radar operation. Messages from onboard positioning systems, on the other hand, can arrive at any random moments in time, which corresponds to the asynchronous mode. It is assumed that both message flows are superimposed on each other. Namely, received in asynchronous mode and recorded as Event Protocols. The program for processing and interpreting measurement

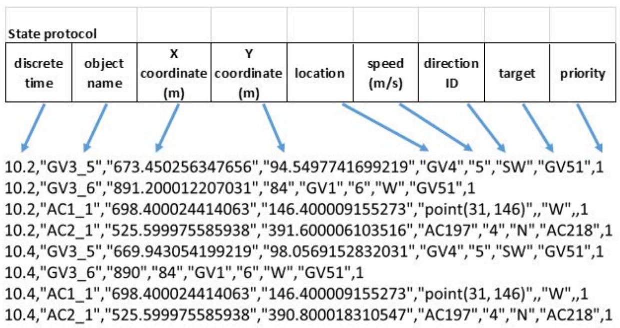

Figure 4. Fragment of a State Protocol file demonstrates the recording format

Event protocol

real-time stamp	object name	X-coordinate (m)	Y-coordinate (m)
10:45:11.82	GV2_2	573,60	315,12
10:45:11.83	AC2_1	525,60	376,68
10:45:11.85	GV2_3	679,95	377,55
10:45:12.00	GV2_2	573,60	318,54
10:45:12.03	AC2_1	525,60	375,46
10:45:12.04	GV2_3	679,46	376,30
10:45:12.23	GV2_3	679,20	372,85
10:45:12.24	AC2_1	525,60	372,24
10:45:12.25	GV2_2	573,60	320,49

State protocol

discrete time	object name	X-coordinate (m)	Y-coordinate (m)	location	speed (m/s)	direction ID	target	priority
10:45:12.00	GV2_2	573,6	318,5	GV108	5	S	GV51	1
10:45:12.00	GV2_3	679,5	377,1	GV28	4	NW	GV51	1
10:45:12.00	AC2_1	525,6	376,0	AC197	4	N	AC218	1
10:45:12.20	GV2_2	573,6	319,5	GV108	5	S	GV51	1
10:45:12.20	GV2_3	679,2	373,0	GV29	5	N	GV51	1
10:45:12.20	AC2_1	525,6	372,4	AC197	4	N	AC218	1
10:45:12.40	GV2_2	573,6	323,3	GV108	5	S	GV51	1
10:45:12.40	GV2_3	679,2	372,0	GV29	5	N	GV51	1
10:45:12.40	AC2_1	525,6	371,6	AC197	4	N	AC218	1

Figure 5. Bidirectional transformation of state and event protocols

data is launched at certain points in time, while data on the location and state of moving objects should be available at the moment the program is launched. The data associated with the start of the dispatcher programs are generated and saved in the form of State Protocols. Figure 5 illustrates the capabilities of the two-way reciprocal transformation of state and event protocols.

The data recorded in the form of Event Protocols and State Protocols are components of the Digital Twin (Saifutdinov et al., 2020; Saifutdinov & Tolujevs, 2021). In this case, it is assumed that Digital Twin performs the functions of only a dynamic data storage, and the solution of a wide variety of monitoring, analysis and control tasks is implemented using the corresponding Applications (Figure 1).

The GTSS program provides a mode to demonstrate processes in the model of the airport transport system, in which the interpretation of data recorded as an external text file is performed. Scenarios specified in the form of "Static positions" and "Dynamic routes" are not used in this mode. In this case, the model is launched using the "read protocol" and "start protocol" keys. The program determines what type of protocol was used to write the file. If the format complies with the State Protocol, then the data is directly interpreted from the protocol lines. If the format conforms to the Event Protocol, then each line is converted to the State Protocol format. To determine the speed and direction of object movement, data from several previous lines of the Event Protocol are used. This operational mode of the model allows one to demonstrate any number of previously simulated and saved scenarios for the purpose of training dispatchers or automatic control programs for airport ground traffic.

Conclusions

The development of GTSS should be considered as the main result of this work. The developed program is focused on working in emulation mode in conjunction with the surface movement control programs at the airport. The models that can be generated with GTSS are radically different from conventional airport transport models that are generated using packages such as TAAM, ArcPORT, SIMMOD and CAST. The main difference between these models is that they are focused on modeling not random, but precisely defined processes, which in this work are referred to as scenarios. Each scenario should lead to the emergence of a critical situation in the transport network of the airport, for which the intervention of the centralized control system is necessary to handle. Currently, the models obtained through GTSS do not yet have software for automatic traffic control, and the corresponding functions of centralized control are performed by the user of the model. In this case, the qualifications of the user must be such that it allows him to create both scenarios: a) a scenario for the occurrence of a typical critical situation and b) a scenario for getting out of this situation.

Vehicle traffic data is recorded in the form of Event Protocols and State Protocols, which are components of the Digital Twin. This data is the only type of information that can be used to test existing or develop new algorithms for centralized surface movement control at an airport. In addition, the simulated protocols can display with any accuracy the real streams of measurement information coming from both SMRs and onboard systems for determining the position of moving objects.

The GTSS program is the first prototype of a simulation software specifically designed to debug automated ground movement control software at an airport. Although the GTSS program was developed using VBA, and the models are created and processed in the MS Excel environment, the functionality of GTSS allows one to perform all the actions described in this work with the model. The GTSS program is universal since it can be used to create a simulation model of the transport network of any airport without writing a new program code. For a complete description of the transport network, it will take 2–3 days for a user to deal only with MS Excel graphical objects and tables. To develop models that can work together with the control software in real time, it will be necessary to develop a new version of the GTSS program, for example, based on the C programming language. Therefore, during the development of the GTSS version demonstrated in this paper, the main goal was to minimize the cost of programming a product that would allow experimental testing of the basic modeling principles focused on scenario processing and data stream generating.

Digital Twin obtained using GTSS is an information base for the development and testing of various applications, ranging from passive monitoring of processes in the airport transport network to fully automatic control of these processes. At the first stage, the tasks of automatic recognition of situations that were determined by the dispatcher as critical should be solved. At the second stage, control programs should be developed that develop specific instructions for the participants in the transport process in order to resolve the critical situation. Sometimes such problems can be solved by developing deterministic rules and presenting them, for example, in the form of decision tables. However, more effective solutions that take advantage of the experience of human experts are currently a hot topic in machine learning research.

Disclosure statement

Authors have no competing financial, professional, and personal interests from other parties.

References

Airport Research Center. (n.d.). *Cast vehicle ground handling*. Retrieved June 9, 2021, from https://arc.de/cast-vehicle-ground-handling/

ArcPORT. (n.d.). *Comprehensive and powerful terminal simulation*. Retrieved June 9, 2021, from http://www.transoftsolutions.com/de/fast-time-airport-capacity-simulation/arcport/

Astarita, V., Giofre, V. P., Festa, D. C., Guido, G., & Vitale, A. (2020). Floating car data adaptive traffic signals: A description of the first real-time experiment with "connected" vehicles. *Electronics*, *9*(1), 1014. https://doi.org/10.3390/electronics9010114

Astarita, V., Guido, G., Vitale, A., & Giofré, V. (2012). A new microsimulation model for the evaluation of traffic safety performances. *European Transport*, *51*(1), 1–16.

Atac. (n.d.). *Simmod PRO! Advanced simulation tool for analyses*. Retrieved June 9, 2021, from https://www.atac.com/simmod-pro.html

Augustyn, S., & Znojek, B. (2015, 28–30 May). The new vision in design of airport. In *International Conference of Scientific Paper AFASES* (Vol. 2, pp. 369–372). Brasov.

Bertino, J., Boyajian, E., & Johnson, N. (2011). 21st century, fast-time airport and airspace modeling analysis with Simmod. *Managing the Skies*, *9*(3), 21–23.

Chen, Y., Shafi, S. Y., & Chen, Y. (16 May, 2020). Simulation pipeline for traffic evacuation in urban areas and emergency traffic management policy improvements. *arXiv:2002.06198v2* [cs.AI]. https://doi.org/10.1016/j.trip.2020.100210

Dimitropoulos, K., Grammalidis, N., Simitopoulos, D., Pavlidou, N., & Strintzis M. (2005). Aircraft detection and tracking using intelligent cameras. In *IEEE International Conference on Image Processing* (pp. 594–597). Genova. https://doi.org/10.1109/ICIP.2005.1530125

EUROCONTROL. (2020). *Specification for Advanced-Surface Movement Guidance and Control System (A-SMGCS) Services* (Ed. 2.0, published 22 April 2020). https://www.eurocontrol.int/sites/default/files/2020-04/eurocontrol-specification-a-smgcs-v-2-0.pdf

Helleboogh, A., Holvoet, T., & Berbers, Y. (2006). Testing AGVs in dynamic warehouse environments. In D. Weyns, H. Van Dyke Parunak, & F. Michel (Eds.), *Environments for Multi-Agent Systems II. E4MAS 2005. Lecture Notes in Computer Science*, *3830*, 270–290. Springer. https://doi.org/10.1007/11678809_16

International Air Transport Association. (2011, February 12). *Vision 2050*. Singapore.

Jeppesen. (n.d.). *Jeppesen total airspace and airport modeler*. Retrieved June 9, 2021, from https://ww2.jeppesen.com/airspace-solutions/total-airspace-and-airport-modeler/

Kaidi, Z. (2019). Design and implementation of a centralized monitoring system for a remote control tower system. In the *IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT)* (pp. 498–503). IEEE. https://doi.org/10.1109/ICCASIT48058.2019.8973147

Law, A. M., & Kelton, W. D. (2000). *Simulation modeling and analysis* (3rd ed.). McGraw-Hill.

Lee, S., Lee, D., Choi, P., & Park, D. (2020). Accuracy-Power Controllable LiDAR sensor system with 3D object recognition for autonomous vehicle. *Sensors, 20*(19), 5706. https://doi.org/10.3390/s20195706

Lopez, P., A., Behrisch, M., Bieker-Walz, L. Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wiessner, E. (2018). Microscopic traffic simulation using SUMO. In *The 21st IEEE Intelligent Transportation Systems Conference (ITSC)* (pp. 2575–2582). Maui, Hi, USA. IEEE. https://doi.org/10.1109/ITSC.2018.8569938

Rintanen, K., & Thomas, A. (2021). *Container terminal automation. A PEMA information paper.* Retrieved June 9, 2021, from https://www.pema.org/wp-content/uploads/downloads/2016/06/PEMA-IP12-Container-Terminal-Automation.pdf

Saifutdinov, F., & Tolujevs, J. (2020a). Analysis of navigation systems for landside transport processes control. In *Reliability and Statistics in Transportation and Communication. Lecture Notes in Network and Systems*, *117*, 552–561. Springer. https://doi.org/10.1007/978-3-030-44610-9

Saifutdinov, F., & Tolujevs, J. (2020b). A model for ground transportation systems simulation at airports under centralized control. *Communications of the ECMS, 34*(1), 5–10. http://www.scs-europe.net/dlib/2020/ecms2020acceptedpapers/0005_is_ecms2020_0016.pdf

Saifutdinov, F., & Tolujevs, J. (2021). Time and space discretization in the digital twin of the airport transport network. *Transport and Telecommunication Journal, 22*(3), 257–265. https://doi.org/10.2478/ttj-2021-0019

Saifutdinov, F., Jackson, I., Tolujevs, J., & Zmanovska, T. (2020). Digital twin as a decision support tool for airport traffic control. In the *61st International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS 2020)* (pp. 1–5). Riga, Latvia. IEEE. https://doi.org/10.1109/ITMS51158.2020.9259294

Spieckermann, S., Stauber, S., & Bleifuß, R. (2012). A case study on simulation and emulation of a new case picking system for a US based wholesaler. In *Proceedings of the 2012 Winter Simulation Conference* (pp. 1490–1501). IEEE. https://doi.org/10.1109/WSC.2012.6465181

Xcelgo. (2020). *Emulation vs simulation – what is the difference?* https://xcelgo.com/emulation-vs-simulation/