# On Microservice Architecture Based Communication Environment for Cycling Map Developing and Maintenance Simulator


Artis Aizstrauts
*Sociotechnical Systems OU*
Tallinn, Estonia
artis.aizstrauts@gmail.com

Dirk Burkhardt
*Darmstadt University of Applied Sciences*
Darmstadt, Germany
dirk.burkhardt@h-da.de

Egils Ginters
*Department of Modelling and Simulation*
*Riga Technical University*
Riga, Latvia
egils.ginters@rtu.lv

Kawa Nazemi
*Darmstadt University of Applied Sciences*
Darmstadt, Germany
kawa.nazemi@h-da.de



***Abstract*—Urban transport infrastructure nowadays involves environmentally friendly modes of transport, the most democratic of which is cycling. Citizens will use bicycles if a reasonably designed cycle path scheme will be provided. Cyclists also need to know the characteristics and load of the planned route before the trip. Prediction can be provided by simulation, but it is often necessary to use heterogeneous and distributed models that require a specific communication environment to ensure interaction. The article describes the easy communication environment that is used to provide microservices communication and data exchange in a bicycle route design and maintenance multi-level simulator.**



***Keywords—simulation, microservice architecture, easy communication environment***


## I. Introduction

The development of processes in society and in the economy is accelerating. The introduction of digital technologies in all industries and in public life is changing the way society lives. On the one hand, the processes are becoming more transparent, but on the other hand, they are more vulnerable and subject to external influences. Development and introduction of technologies is non-linear, since it is determined not only by the quality of technology, but exclusively by social demand. The predictable result of management is determined by many random factors and the interaction between technical and social components.

Certain corrections are made by changes in the environment. Today, almost any really functioning system is sociotechnical, where technologies and social environment are inseparable. Modelling such systems is complex and expensive. The use of analytical methods leads to serious simplifications in describing the functioning of the system. Simulation is used to save funding and achieve a more transparent and flexible result. Simulation allows to play around different development scenarios and makes it easier to design software, since the notation for describing the simulation models takes a higher level of abstraction and becomes closer to the perceptions of developers. Basically, simulation is used to identify a development trend or forecast but is rarely used to calculate an accurate result.

Since modern systems are heterogeneous, so are simulation models. To achieve the result, it is necessary to interconnect several different technologies of simulation, data analytics [1] and a set of distributed software modules. There are well-developed and widely used methods of providing communication and interaction of distributed simulation models, for example, HLA [2].

However, in this case, one cannot do without special knowledge in programming, which causes additional problems and financial costs for stakeholders. The communication environment presented by the authors in this article is more accessible for modelers and provides the necessary flexibility to implement rapid changes in the architecture of the simulator.

The modern city and its transport system are a classic example of a complex sociotechnical system with a multi-level and distributed architecture. Considering the functioning as the interaction of engineering and social components can stimulate an increase in the productivity of planning and system safety [3].

The authors of this article have been working on the VeloRouter cycling map simulator in recent years. The designing was based on findings of previous EU FP7 project FUPOL [1]. VeloRouter users are municipal policy planners and citizens, as it serves both infrastructure design and daily travel planning.

The purpose of this article is to explain the fundamentals of the Easy Communication Environment (ECE) and to show its application to ensure the functioning of VeloRouter [5].

The following sections outline the basic assumptions, concepts and functionality of both VeloRouter and the proposed communication mechanisms based on the microservices architecture.

## II. VeloRouter – A Simulator for Planning and Operating a Network of Bike Paths

Today, more than half of the world's population lives in cities, with steady growth trends [6].

The rapid spread and development of technologies and social media has led to the development of areas such as urban analytics and participatory planning. Hence, more thoughtful decisions can be made and promoted.

Four main urban planning objectives can be highlighted [7]:

- *Lively cities* are achieved when citizens in the urban space are free to use certain transport routes and, with the help of bicycles and similar means of transportation, acquire the possibilities of free movement and communication.


978-1-7281-9105-8/20/$31.00 ©2020 IEEE

- *Safety* is achieved by reducing the number of necessary heavy urban transport and the density of the coverage.

- *Sustainability* is encouraged through "green mobility", that is, walking, cycling or more environment friendly public transport.

- Reducing environmental pollution improves the *city's health*. To achieve this is necessary to promote movement on foot or by vehicles that do not use fossil fuels.

The above basic principles determine the increase in the share of green transport. One such means of transport is a bicycle, the use of which requires adequate infrastructure. The tracks can also be used by scooters and skaters. The cycling route map must be complete and ensure movement throughout the city, respecting the needs of the population. However, financial constraints do not allow the construction of quality roads in any part of the city. To respect these limitations, cyclists must be directly involved in the planning process (see Fig. 1).

the impact of the developing map of new routes. Existing tools such as petitions.net, bikemap.net or mapmyride.com cannot solve the above complex problems [5].

VeloRouter is a tool for the designing and operation of urban bike paths, based on the participation of the population and other stakeholders [8] (see Fig. 1).

There are three main stakeholder groups: municipalities, infrastructure developers and cyclists. The provided advantages of VeloRouter are as follows:

- The statistics section of VeloRouter allows to calculate and analyze the most popular travel routes. *Municipal professionals* can get feedback from the citizens and offer a new cycle path construction.

- *Developers* can use the digital twin of the cycling infrastructure to predict depreciation and timely plan the repair work.

- *Cyclists* can proactively predict the load on the route and participate in infrastructure development by presenting and discussing their own and other projects.

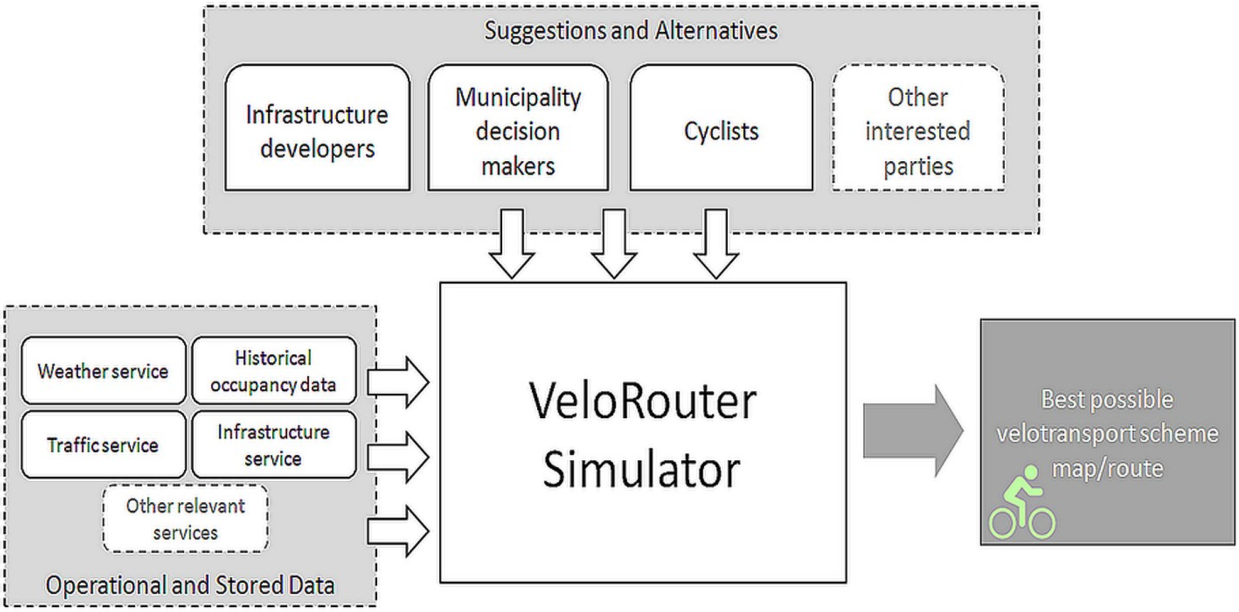


Fig. 1. VeloRouter: Conceptual model of a system for planning and operating urban bike paths and routes.

In order to increase the share of cycling, the infrastructure must be of enough quality. Possible routes must be connected and well-lit at night. Required attributes are bicycle / scooter rental, repair services and monitoring and coordination of the transport network. For a successful trip, proactive route planning must be ensured, which must respect the quality and terrain of each segment. To temporarily plan the arrival at the destination of the trip, it is necessary to predict the load of certain sections of the route on different days of the week and weather conditions.

At the same time, citizens do not have the appropriate tools to effectively interact with decision-makers. There is no feedback on the condition of the existing infrastructure and

The VeloRouter simulator uses four basic data sets: weather data, cycling route load data, traffic forecasts, and infrastructure condition data.

Weather data consists of historical data and forecasts for a possible trip. The cyclist can determine which weather conditions are suitable for each of them. This helps to predict the possible load on the route, that is, it is possible to find out how many cyclists will be on one segment or route at a time. The traffic service provides a forecast of traffic congestion on the streets. The infrastructure data set includes information on the technical characteristics and actual condition of each cycle path. OpenStreetMap spatial data is used to build the road network.

VeloRouter simulator has a multi-level heterogeneous architecture. Although agent-based simulation in Repast Symphony is mainly used, data exchange and synchronization with discrete-event simulation must be ensured. Sub-models' collaboration is implemented based on microservices interaction.

## III. Easy Communication Environment

VeloRouter simulator is built on a microservice ECE architecture. The communication mechanism of the micro-service is based on the Advanced Message Queuing Protocol (AMQP), which ensures the interaction of the elements of the application set. In turn, the harmonization of data formats is implemented by assistance of JavaScript Object Notation (JSON). Such basic principles ensure the resilience of the implemented solutions, that is, it is possible to replace damaged and occupied micro-services in critical situations. The implementation becomes independent of the programming platform, because only the interface is important. This facilitates the addition of functionality by the introduction of new features. There is an opportunity to easily create new links and interactions between sub-models by designing larger network architectures [9].

The ECE communication mechanism is specified in A, B, C, and D lanes (see Fig. 2).

*1) A - ECE presentation lane.* The essence of the presentation lane is similar to the sixth level protocol of the ISO-7498 open systems Basic Reference Model. The main task of A-lane is to harmonize data formats to ensure the exchange of sub-models messages. The JSON language and a harmonized data exchange message format are used. The procedure for exchanging messages is determined by the AMQP protocol.

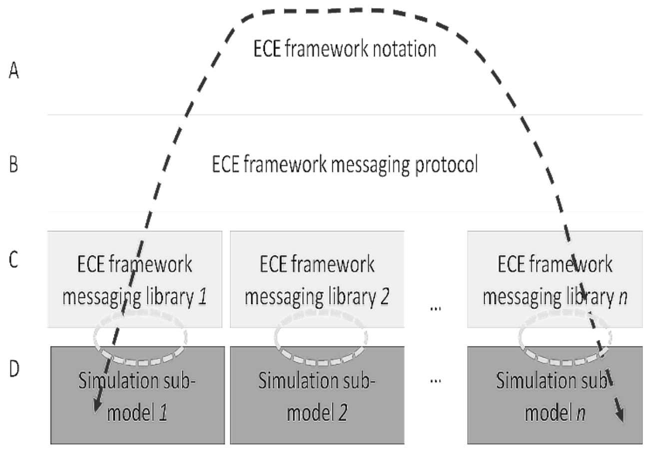


Fig. 2. Stratification of the Easy Communication Environment.

*2) B - ECE messaging control and transportation lane.* The protocol is responsible for transmitting messages in the microservice architecture between the individual sub-models that make up the simulator's task network.

The lower lane set of the ECE stratification within the meaning of ISO-7498 describes the transport station.

The ECE model has a minimal set of standard messages. The communication scheme is shown in Fig. 3 and Fig. 4. Any interaction begins with sending a Request message to the respondent's sub-model.

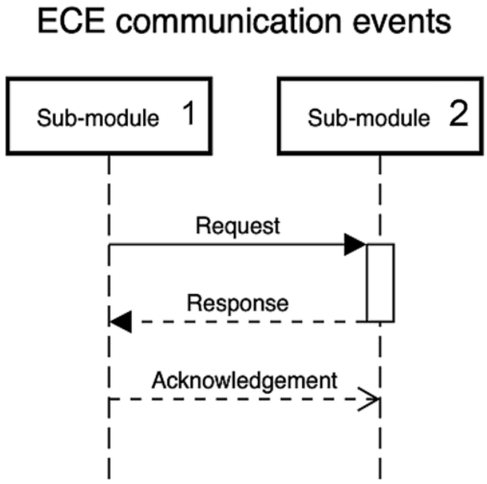


Fig. 3. The sequence of communications in ECE.


```
{
   "type": "request",
   "request_guid": "2fbec59a-7686-4f96-9686-88aaff7ef3a5",
   "service_guid": "9b0ae71b-aeb6-42db-bbcf-2875cafb0a39",
   "service_endpoint": "closes_path",
   "payload": {
      "location_node_start": 847584,
      "location_node_end": 234903
   }
}
```


Fig. 4. Formation of the message Request.

Each simulation sub-model works in parallel and autonomously until it receives a Request from another sub-model. The interrupt called by the request is processed and the respondent sub-model sends a Response message. To consider a session established, the initiator responds with an Acknowledgment message. In this way, an asynchronous handshake procedure is implemented.

For the successful implementation of two or more sub-model interactions, all lanes of the ECE communication model must be involved. Fig. 5 shows in detail the lane cooperation to ensure the exchange of messages between two objects in the ECE environment.

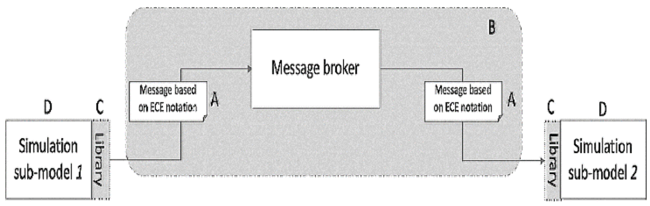


Fig. 5. Message broker use for interaction between the two sub-models.

The message transport and monitoring protocol AMQP [10] provides message registration and queue processing, as well as message routing in the ECE environment. One of the reasons for using AMQP is the high delivery confidence and resilience of the mechanism [11]. Confidence is considered appropriate if the mechanism complies with the procedures set out in the second level of quality of service (QoS) [12].

The basic level allows to save resources of communication session time, however, does not guarantee safe and reliable communication. The message is sent but not acknowledged [13]. Such a mechanism is possible if the messages are insignificant and informative, such as announcements or advices, but this approach is not applicable if each message initiates actions that are important for the functioning of the system. Such a mechanism could be used in broadcasting systems, but it would not be useful to invest resources in sending a message to a specific respondent without asking him to reply.
Although the second level of QoS is considered sufficient in the ECE ideology, there are serious doubts as to whether such a mechanism can be used in simulation models that need to work in real time, such as automatic guided vehicle positioning on the route.

*3) C – ECE messaging library.* The main task of the library is to harmonize the simulation model notation with JSON and AMQP to ensure interaction with the brokerage mechanism. It is the broker who is responsible for delivering each message to the correct respondent.

*4) D - Sub-models.* Sub-models provide the implementation of specific simulation algorithms, whose common interaction determines the overall simulation result. Distributed simulators are built in this way [15]. With this approach, sub-models are allocated to specific domains, i.e. load balance management, parallelization of the modeling process, etc. Communication is critical to maintaining distributed modeling and achieving the primary goal of modeling.

The ECE communication model in the VeloRouter simulator is used to ensure cooperation between both simulation sub-models and real services. The cyclist has proactively prepared the planned travel route in appropriate weather conditions, if the load and quality of the bicycle paths meet the wishes of the group of riders. However, road repairs may take place at any time, as well as changes in weather conditions, which cause changes in the load and quality of the previously simulated route. In this case, the load simulation sub-model interacts with other VeloRouter services using both historical and on-line data, while the ECE mechanism provides simulator services interaction.

The above approach ensures the reliability and transparency of data exchange between respondents of the ECE architecture.

## IV. CONCLUSIONS

Designing and operating a multimodal and sustainable urban transport system is challenging. The use of simulation helps to achieve the desired result. However, the heterogeneous structure of tasks determines the distributed architecture of the simulator, which also consists of several interacting sub-models.

The communication environment of the VeloRouter - urban cycling modeling and operation system is built based on microservices. Microservices allow to separate the performance of different functions, as well as facilitate maintenance and prototyping. To ensure distributed simulation sub-models interaction reliability and scalability the ECE communication and data exchange mechanism is based on JSON data formats and the AMQP protocol. The present architecture can subsequently be used to build communication systems for other distributed simulators, the operation and modification of which are implemented without the participation of software engineers.

## ACKNOWLEDGMENT

The article publication is initiated by FP7 FLAG-ERA FuturICT 2.0 (2017-2020) "Large scale experiments and simulations for the second generation of FuturICT" STIMUL project "Intermodal bicycles network as efficient kind of urban green transport (VeloRouter 2.0)".

## REFERENCES

[1] I. Sitova, and J. Pecerska, “Process Mining Techniques in Simulation Model Adequacy Assessment”, in *Proc. of 60th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS 2019)*. ISBN 978-1-7281-5710-8, IEEE, pp. 49-53, 2019.

[2] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)— Object Model Template (OMT) Specification. ISBN 978-0-7381-6249-2, IEEE Computer Society, 2010.

[3] L. J. Hettinger, A. Kirlik, Y. M Goh, and P. Buckle, “Modelling and Simulation of Complex Sociotechnical Systems: Envisioning and Analysing Work Environments”, *Ergonomics*, 58(4), pp. 600-614, 2015.

[4] E. Ginters, A. Aizstrauts, G. Dreija, M. Ablazevica, S. Stepucev, I. Sakne, M. Baltruks, M. A. Piera Eroles, R. Buil, M. Gusev, and G. Velkoski, "Skopje Bicycle Inter-modality Simulator – E-involvement Through Simulation and Ticketing", in *Proc. of 26th European Modelling & Simulation Symposium (EMSS 2014)*. ISBN 978-88-97999-38-6, Bordeaux, pp. 557-563, 2014.

[5] E. Ginters, M. Baltruks, I. Sakne, and J. Merkurjevs, “Dual Use Bicycle Path Network Designing and Exploitation Environment – VeloRouter”, in *Proc. of European Modeling and Simulation Symposium (EMSS 2016)*, Cyprus, pp. 10-14, 2016.

[6] United Nations Population Division. World Urbanization Prospects: 2018 Revision, 2019. Available; https://data.worldbank.org/indicator/SP.URB.TOTL.in.zs. [Accessed: October 27, 2019].

[7] I. Gehl, “Cities or People”, Island Press, 2010.

[8] E. Ginters, M. Mezitis, and D. Aizstrauta, "Sustainability Simulation and Assessment of Bicycle Network Design and Maintenance Environment", in *Proc. of the IEEE 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*. ISBN 978-153866477-3, Mauritius, pp. 1-7, 2018.

[9] M. Amundsen, M. McLarty, R. Mitra, and I. Nadareishvili, “Microservice Architecture: Aligning Principles, Practices, and Culture”, O'Reilly Media, 2016.

[10] S. Vinoski, “Advanced Message Queuing Protocol”, IEEE Internet Computing, pp. 87–89, 2006.

[11] OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0, 2017. Available: http://docs.oasis-open.org/amqp/core/v1.0/. [Accessed: October 25, 2019].

[12] N. Naik, “Choice of Effective Messaging Protocols for IoT Systems: MQTT CoAP AMQP and HTTP”, in *Proc. IEEE Int. Syst. Eng. Symp. (ISSE)*, pp. 1-7, 2017.

[13] Message Quality of Service (QOS), 2019. Available: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.1.0/com.ibm.mq.dev.doc/q123795_.htm#q123795___qos. [Accessed: October 25, 2019].

[14] JavaScript Object Notation, 2019. Available: https://www.json.org. [Accessed: October 25, 2019].

[15] A. Aizstrauts, E. Ginters, M. Baltruks, and M. Gusev, “Architecture for Distributed Simulation Environment”, *Procedia Computer Science*, vol. 43, pp. 18-26, 2014.