

Support of Event-Graph Lectures by the MMT E-learning System

Martin Bicher^{1,*}, Gasper Music², Irene Hafner³, Felix Breiteneker¹

¹Inst. of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *martin.bicher@tuwien.ac.at

²Faculty of Electrical Engineering, University of Ljubljana, Slovenia

³dwh Simulation Services Vienna, Austria

SNE Simulation Notes Europe SNE 24(1), 2014, 47 - 50

DOI: 10.11128/sne.24.en.102_33

Received: Dec. 10, 2013 (Selected ERK 2013 Postconf. Publ.);

Accepted: February 20, 2014;

Abstract. In a lot of technical professions nowadays an advanced education in mathematical modelling and simulation has already become a prerequisite. Therefore universities are required to provide high quality lectures and teaching materials to guarantee the qualification of their graduates within this field of study. For engineering students surely the emphasis has to be laid on discrete-event modelling, as the understanding of event-operation and scheduling provides the base for designing management and control strategies. Often the concept of event-graph modelling is used to teach the basis of discrete-event simulation in a graphical way. The paper shall give an idea on how methods of event-graph modelling can be taught to students using blended learning concepts. In this special case the e-learning platform MMT (Mathematics, Modelling and Tools) is used, a kind of virtual laboratory for modelling and simulation. The basic ideas of event-graph modelling, how event-graph based examples are prepared for the e-learning platform and the basic properties of the MMT-server are furthermore presented in this paper as well as some simple examples are shown in addition.

Introduction

In a lot of technical professions nowadays an advanced education in mathematical modelling and simulation has already become a prerequisite. Therefore universities are required to provide high quality lectures and teaching materials to guarantee the qualification of their graduates within this field of study.

For engineering students surely the emphasis has to be laid on discrete-event modelling, as the understanding of event-operation and scheduling provides the base for designing management and control strategies.

Regarding discrete-event simulation one of the most challenging parts of the modelling process is the formal description of the model, which is on the one hand essential for the implementation and on the other hand necessary for a comprehensible documentation of the modelling process. One of the most figurative description forms for these kind of models is the so called event-graph modelling, formally described in Section 1. This kind of representation strategy is extremely suitable for queuing processes.

Although event-graphs are very simple to understand they are yet very flexible and really complex behaviour can be observed. They are suitable for deterministic as well as stochastic models.

In order to support lectures dealing with event-graphs targeted e- or blended-learning should be used. Common e-learning platforms usually provide the opportunity to offer teaching material and sometimes create tests with multiple-choice or numerical questions, where students can proof their knowledge. However, regarding the pictorial representation and the complex and maybe stochastic behaviour of event-graph models, these opportunities might not be sufficient. Therefore the blended-learning system MMT (*Mathematics Modelling and Tools*) system, in detail explained in Section 2, was developed by dwh GmbH Vienna. The platform is based on the idea, that students can additionally experiment on pre-implemented, accurately tested and high quality simulation examples, by changing certain parameters in a browser window. The model is calculated on a fast server in behind and the results are presented either in textual, image or video form directly within the browser window.

In order to integrate user friendly event-graph examples to the MMT server, a new event-graph library was developed in MATLAB.

1 Bases of Event-Graph Modelling

Aforementioned event-graphs are probably the most popular ways to describe a discrete-event model in figurative form. Yet, those models have to fulfil the following conditions in order to be described by an event-graph:

1. Only a finite number of events may occur during simulation-time.
2. All observed state variables remain constant during the time between each two sequential events. These variables change their value only directly at the events.
3. The model is dynamical respectively time-dependent.
4. The time when an event takes place can depend on parameters, random variables and the state variables itself. Therefore the events can be triggered by a so called event-list, which is updated each time the variables change.

The first two conditions are based on fundamental concepts for discrete-event simulation. Usually the set of events can be clustered by condition of occurrence and change of the state variable. Events caused by the same conditions and causing the same change of the state variable are hereinafter called event-type.

Satisfying the aforementioned conditions a discrete-event model is defined by a finite set of possible event-types, linked to conditions when they occur and how they change the state variables, and a finite set of state variables with corresponding initial conditions at $t = t_0$. These ideas inspire defining so called nodes, usually represented by circles - one for each of the event-types. The impact of the event-type on the state variable is quoted usually directly below the node. The causal relationship between the event-types, necessary to consequently trigger an event, is defined by so called edges, represented by connecting arrows. Conditions for the occurrence of an event are quoted directly above the arrow. These might be e.g. time delays and or causal conditions. Before we introduce an example the formal description of an event graph (see also [4]).

1.1 Event-graph formalism

During the following formal description we mainly use the notation of Yucesan and Schruben 1992 [3]. Therefore a quadruple

$G = (V(G), E_s(G), E_c(G), \Psi_G)$ is called event-graph or simulation-graph if...

- $V(G) = \{E_1, \dots, E_k\}$ is a finite set of event-types (called event-vertices),
- $E_s(G) = \{s_1, \dots, s_{n_s}\}$ is a set of scheduling edges,
- $E_c(G) = \{c_1, \dots, c_{n_c}\}$ is a set of cancelling edges (For modelling reasons $E_s(G) \cap E_c(G) = \emptyset$ holds. The set $E(G) := E_s(G) \dot{\cup} E_c(G)$ shall be called the set of all edges of the event graph.),
- $\Psi_G : E(G) \rightarrow V(G) \times V(G)$ is a, not necessarily injective, function, called incidence function, assigning an ordered pair of vertices to a given edge.

The edges, each linked to two vertices (or event-types) by the incidence function, are responsible for the causal relationship between those two events. They denote if, how and when the second of the two events is scheduled (or cancelled from the schedule) after the occurrence of the first one. So far the basic construct of the event-graph is defined which does not contain any definition how state changes take place.

Let $\Omega \in \mathbb{R}$ be the so called state-space, containing all possible states of the model. Furthermore a model M based on an event-graph G is called event-graph model if the following Functions seen in Table 1 are defined and used. Finally the seven-tuple $M = (\mathcal{F}, \mathcal{C}, \mathcal{T}, \Gamma, \mathcal{P}, \mathcal{A}, G)$ is called event-graph model. It is well-defined and can be simulated in a unique way by any event-graph simulation software.

Function	Use
$\mathcal{F} : V(G) \times \Omega \rightarrow \Omega$	Specifies the state change for each event-type.
$\mathcal{C} : E(G) \times \Omega \rightarrow \{0, 1\}$	Specifies the condition of the edge.
$\mathcal{T} : E(G) \times \Omega \rightarrow \mathbb{R}^+$	Specifies the delay time of the edge.
$\Gamma : E(G) \times \Omega \rightarrow \mathbb{R}^+$	Specifies priorities regarding execution of the edge.
$\mathcal{P} : V(G) \times \Omega \rightarrow \mathbb{R}^+$	A function defining event parameters.
$\mathcal{A} : E(G) \times \Omega \rightarrow \mathbb{R}^+$	A function defining edge attributes.

Table 1: Functions for an Event-Graph Model.

Surely this completely theoretical is very difficult to understand. Therefore, similar to most simulation techniques, the idea of event-graphs is best introduced studying examples.

1.2 Multiple server queue

Figure1 shows how a part of how the browser screen looks like, when opening the MMT-example 'Multiple Server Queue'. It deals with an implementation of a classical model, perfectly suitable for event-graph novices. The model deals with a queue of objects focussed on the length of the queue. Directly after the queuing process the objects are treated by one of k different servers. There are many ways to interpret this model with real life aspects. One can think of ...

- ... goods waiting to be processed by one of k -robots.
- ... data packages waiting for being handled by a k -kernel processor.
- ... customers waiting in a shopping mall in front of k cash desks.

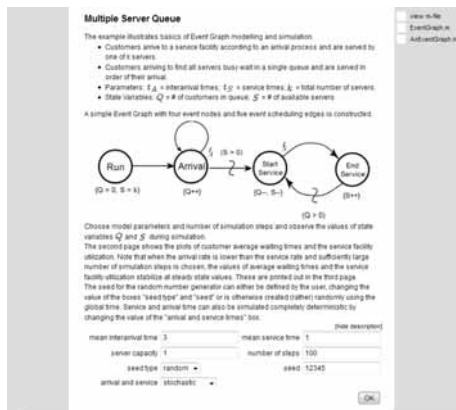


Figure 1: Browser screen of the MMT showing the multiple server queue example.

In all cases the same set of parameters is required:

1. a number of servers k
2. a rate of incoming objects t_A
3. a service time t_S
4. an initial set-up

We furthermore focus on the event-graph illustration in the centre of Figure1 and explain the ideas of this event-graph. First of all the set of all possible event-types is

given by

$$V(G) = \{\text{Arrival, Start service, End service}\}$$

as between those events the queue is idle. Initially by convention also always a so called 'Run' event is scheduled necessary to start the simulation. In this simple example the positions, orientations and conditions of the scheduling edges are nearly self explanatory if one thinks about the causal relationships between the event-types.

The lower section of the browser window seen in Figure1 is reserved for experimenting with input parameters for the simulation. In this case the modelling type can be changed from deterministic (i.e.deterministic arrival and service time) to stochastic using $t_A \sim Exp(\text{mean}(t_A))$ and $t_S \sim Exp(\text{mean}(t_S))$. The exponentially distributed random numbers are gained by a random number generator and can either be calculated by a 'random' seed (dependent on the global time), or by a user defined seed.

Pressing the 'ok' button, in this case, MATLAB is started in the background calculating the multiple server queue with the chosen input parameters. An output example plot is shown in Figure2. Arrival time (mean 1) and the service time for 8 servers (mean 10) are distributed exponentially as well. Before going into specific code details a closer look at the MMT server is taken.

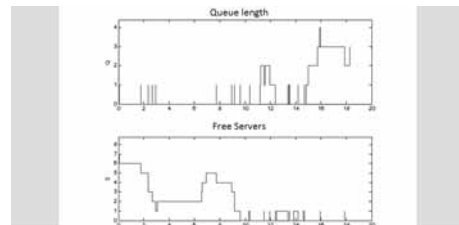


Figure 2: Results of the multiple-server queue. Length of the queue can be seen as well as the occupancy of the servers.

2 MMT Server

The MMT-server was developed and is maintained by dwh-Vienna and is mainly used for lectures at the Vienna University of Technology. Additionally it is used within external courses for modelling and simulation

too. The web-interface provides the basis for high quality e-learning. Students as well as lecturers benefit. The most important properties of the server are presented here (see also [1] and [2]).

2.1 Virtual laboratory

The MMT system provides the opportunity to upload MATLAB, Simulink (Simscape, Simmechanics), Java and AnyLogic models to create a virtual laboratory. The models can be called from the internet-browser, simulation takes place in the background and the results are then sent back to the browser window again in text, picture and/or video form. Specified parameters for the models can be changed directly within the MMT-page by changing the values in provided HTML text-boxes. Thus programming skills are not required for experimenting with high quality modelling and simulation examples. Currently about 500 MATLAB, 50 Simulink, and 30 Java/Anylogic models can be tested and experimented. Examples written within the free-ware programming languages R and Octave are planned to be included too.

2.2 Upload download of files

As other e-learning systems, also MMT provides the opportunity to offer lecture notes, images or pdf explanations for download. Using an unique node-based course-oriented administration system grants, that each participating student gets access only to those files intended for her/him.

2.3 Improve programming skills

As programming skills are becoming more and more important within technical professions the MMT server provides the opportunity downloading each source code, used for the models, and manipulate it at the home-pc. The administrators of the server work hard to guarantee that every code fulfils high quality standards and is fully commented.

3 Summary and Outlook

Regarding examples like the multiple server queue, the event-graph library improves the usage of the MMT server, as discrete event simulation is a very important topic within the area of control design modelling. In collaboration with the University of Ljubljana

the Vienna University of Technology several other event-graph examples were implemented during the last years, which were in consequence uploaded to the MMT system increasing the teaching value of the platform. Currently we plan to extend the usage of the MMT system in order to be able to simulate more serious discrete event models e.g. by including libraries for treating DEVS formalism.

Acknowledgment

This contribution is a post-conference publication from ERK 2013 Conference (22nd International Electrotechnical and Computer Science Conference, September 16 - 18, 2013, Portorož, Slovenia), with sessions organized by SLOSIM, the Slovenian simulation society (EUROSIM member). The contribution is a modified publication from the paper published in the Proceedings of ERK 2013, [5].

References

- [1] Hafner I, Bicher M, Winkler S, Fitsch U. MMT - An E-Learning System based on Computer Numeric System for teaching Mathematics and Modelling. In Troch I, Breitenecker F, editors. Preprints MATHMOD 2012 Vienna - Full Paper Volume. *MATHMOD 2012* 2012 Feb; Vienna. Vienna: ARGESIM and ASIM.
- [2] Hafner I, Bicher M, Peterseil T, Winkler S, Fitsch U, Nagele-Wild N, Wild W, Breitenecker F. Achievements in result visualization with the computer numeric e-learning system mmt. In Troch I, Breitenecker F, editors. Proceedings of the European Modeling and Simulation Symposium, 2012. *THE 24TH EUROPEAN MODELLING & SIMULATION SYMPOSIUM 2012* Sept; Vienna. Genoa, Italy: ARGESIM and ASIM.
- [3] Yücesan E, Schruben L. Structural and behavioral equivalence of simulation models. *ACM Trans. Model. Comput. Simul.* 1992. doi: 10.1145/132277.132281.
- [4] Chick S, Sánchez P, Ferrin D, Morrice D, Schruben L, Roeder T, Chan W, Freimer M. ADVANCED EVENT SCHEDULING METHODOLOGY
- [5] Bicher M, Music G, Hafner I, Breitenecker F. Support of Event-Graph Modeling by the MMT E-learning System. In: Zajc B, Trost A, editors. ERK 2013. Proceedings of the 22nd International Electrotechnical and Computer Science Conference. 2013 Sept; Portorož; Ljubljana: Slovenia Section IEEE. P. 161-164.