EVENT SCHEDULING MADE EASY: BASIC SIMULATION FACILITY REVISITED

Luís S Dias [a], Guilherme A B Pereira [b], José A Oliveira [c]

[a][b][c] Departamento de Produção e Sistemas
Universidade do Minho
4710-057 Braga – Portugal

[a] lsd@dps.uminho.pt, [b] gui@dps.uminho.pt, [c] zan@dps.uminho.pt

ABSTRACT
In this paper, the use of Event Scheduling concepts for modeler-client communication, acting as an automatic generator of simulation programs under event scheduling paradigm - Basic Simulation Facility, thus eliminating any programming effort and expertise, is studied.

The main idea behind this work is to enhance the utilization of flowcharts in modeling, making it a great contribution to automatic generation of simulation programs, keeping it simple and portable.

The software tool developed is supposed to perform a sequence of what-if questions with a simple interaction with the user. These questions are then capable of automatically generating a program structure based on Event Scheduling simulation approach, using the Basic Simulation Facility routines.

This software tool would then contribute to the generalization and better understanding of the use of simulation as it only requires expertise in a basic simulation approach – Event Scheduling, thus incorporating simple flowcharts defining the system and its functioning rules.

Keywords: Event Scheduling, Simulation, Basic Simulation Facility, Automatic Generation of Simulation Programs

1. INTRODUCTION

Simulation is simply the use of a computer model to "mimic" the behavior of a complicated system and thereby gain insight into the performance of that system under a variety of circumstances (Thesen and Travis 1990).

In this paper, we keep the suggestion of using a (simple) graphical support as a representation of how the system really behaves, but these diagrams will also act as the source to the automatic generation of simulation programs.

In particular, "…Discrete event simulation models are run by tracing events over time, particularly those events that change the state of the system. Since we do not normally think of systems in terms of events and state changes, we usually use a simulation language or software package that allows us to represent the model more naturally. The computer then translates this to an event oriented approach (i.e., events and state changes) to actually run the model…" (Thesen and Travis 1991).

The traditional approach for discrete event simulation modeling (Dias and Rodrigues 2002), includes visual support diagrams for modeler-client communication purposes (model interpretation and validation) and also to act as the basis for simulation language program construction.

Michael Pidd (1992), and Tocher (1963), even support that when generic programming languages were replaced by specific purpose simulation languages the use of paper diagrams remained as a previous step to programming.

The use of visual support diagrams to help the programming step of a simulation project is very common – this paper emphasizes this step, by proposing a way of automatically translating it into a simulation program. In fact, the abstraction of these diagrams serve as a support to the communication between the simulation client and the modeler (simulation expert), but also help the construction of the corresponding computational programs (Clementson 1982).

Dias and Rodrigues (2002), support that the new powerful graphical interfaces available with modern simulation languages (Dias, Pereira and Rodrigues 2007) are clearly programmer oriented, raising the difficulty in communicating with the client and still requiring enormous simulation expertise to use them.

The use of Event Scheduling concepts for modeler-client communication, acting as an automatic generator of simulation programs under event scheduling paradigm - Basic Simulation Facility (Thesen 1978), thus eliminating any programming effort and expertise, is studied.

The main idea of this work is to enhance the utilization of flowcharts in modeling, making it a great contribution to automatic generation of simulation programs, keeping it simple and portable.

Flowcharts are probably the former and most widely used graphical syntax in behavior specification (Gilbreth and Gilbreth 1921). The first known mathematical formalization was made by Nassi and Shneiderman (1973). It can be accepted as a universal visual language, and it is easy to assume that every professional, in some technical work, has already used it.

2. SOFTWARE TOOL DEVELOPED

The work presented in this paper could constitute a major step towards the generalization of the use of simulation. In fact, we suggest the use of a simple interface (Event Scheduling Flowcharts) to model a real situation. Then we present a tool capable of generating a simulation program. Based on event scheduling simulation modeling philosophy, our tool automatically generates a program to use Basic Simulation Facility routines. Furthermore the mentioned automatic generation of simulation programs does not require great expertise in simulation.

The Event Scheduling simulation philosophy is based upon the identification of events. An event corresponds to a point in time where there is potentially a modification on the state of the system under analysis. The identification of each event is complemented with the definition of the tasks to be performed each time an event occurs. These tasks would include:

1. Managing queues (either removing or inserting entities in queues)
2. Managing resources utilization (either seizing or releasing resources)
3. Recording statistics (for future evaluation of performance indexes, i.e., average waiting time in queue, average queue length, average resource utilization, etc.)
4. Generating random variables
5. Managing future events schedule

The definition of events above is very broad, promoting the construction of general models, hence promoting the automatic generation of simulation programs.

In fact, this broad explanation of an event and the above definition of tasks involved in each event made it possible to develop a set of general routines that apply to a large set of real problems, hence making adequate the use of this facility as far as the Event Scheduling approach is used. One of these facilities is the Basic Simulation Facility – BSF (Thesen 1978).

BSF includes four routines:

1. INIT – essentially dedicated to the design and initialization of the data structure that implements the simulation
2. INSERT – basically dedicated to the insertion of a record into a file (could be the implementation of an arrival to a queue, or seizing a resource or even the planning of a future event)
3. REMOVE – basically dedicated to the removal of a record from a file (e.g. the removal of an entity from a queue, or releasing of a resource or even getting information of a future event)
4. REPORT – essentially dedicated to the computation performance measures

These routines and the philosophy associated could be found (implemented) in various programming languages. Nevertheless, it is essential to develop a computer program, specifically dedicated to the system under analysis, that would invoke these routines, thus creating a mimic of the system.

This step is therefore a step that implies expertise in Event Scheduling simulation philosophy. This paper is then dedicated to present a tool that, based on the Flowcharts constructed for each of the events identified and based on a simple questionnaire related to some particularities of the Flowcharts, would automatically generate a program specifically developed to respond to a real system.

The software tool developed is supposed to perform a sequence of what-if questions with a simple interaction with the user. These questions are then capable of automatically generating a program structure based on Event Scheduling simulation approach, using the Basic Simulation Facility routines mentioned above.

The dialog with the user is based on four possible options:

1. Add Event
2. Remove Event
3. List Events
4. Compile

"Add Event" is the option responsible for adding an event to the model. The corresponding dialog incorporates knowledge on the type of event (whether corresponding to an arrival event or an event related to the use of a resource); on the random variable associated; on the number of resources available, if applicable; on the following resource used, if any, and also on the conditions that would permit that use.

"Add Event" is really the most important option. Here, we demonstrated what exact questions would simultaneously completely define each event, i.e., which questions would be sufficiently self-explanatory to fully define the important information related to each event and also integrate (coordinate) the follow-up, if any, to the use of the next resource.

"Remove Event" and "List Events" just intend to implement the basic options of respectively removing a previous defined event or listing events entered. The "Compile" option simply generates the program according to the answers to the what-if questions that really represent the system rules that will cause the mimic of the system and then compiles the program, enabling its execution. This phase would also create a text file as a result of the mentioned dialog with the user. This particularity of the software tool also permits that a user, well knowing the features of this file, could then simply carry out the previous definition of that file – this would enable to automatically generate the Basic Simulation Facility program without the necessity of executing the mentioned dialog. This represents an operation that requires a certain expertise in Event Scheduling philosophy but, nevertheless, the dialog is always a way to achieve the same result.

3. APPLICATION EXAMPLE

For a brief explanation of the tool developed, we present a simple system which could be represented by a simple simulation model following Event Scheduling approach.

The system incorporates the arrival of entities, according to time between arrivals described by an exponential type of distribution with mean 3 minutes.

Once in the system, the entities would require the use of a first resource (Resource_1). The time for Resource_1 utilization would follow a Random Distribution between 5 and 9 minutes. Then, with a discrete type of distribution, the entity would also require the use of a second resource (Resource_2) with an associated probability of 80%. The time for Resource_2 utilization would follow a Normal Distribution with mean 4 minutes and a standard deviation of 0,25 minutes.

There is a first-in first-out type of queue associated with each resource utilization.

The entity will then leave the system.

The Event Scheduling philosophy would identify three events – Entity Arrival Event (Figure 1); End of Resource_1 Utilization (Figure 2); End of Resource_2 Utilization (Figure 3).

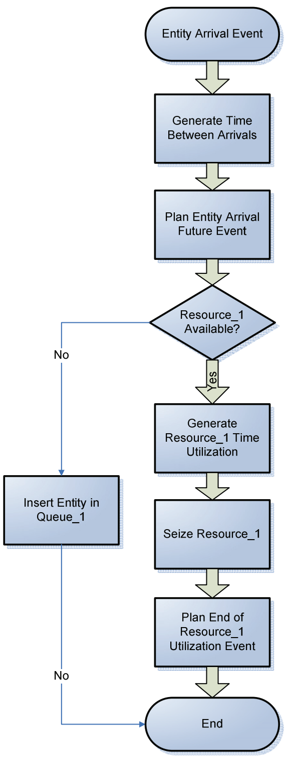

Figure 1 – Entity Arrival Event Flowchart

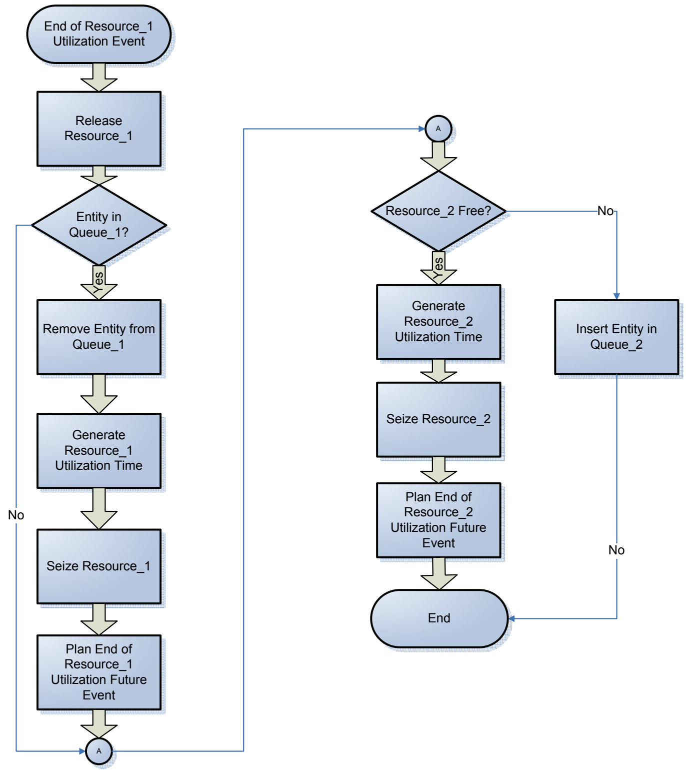

Figure 2 - End of Resource_1 Utilization Event Flowchart

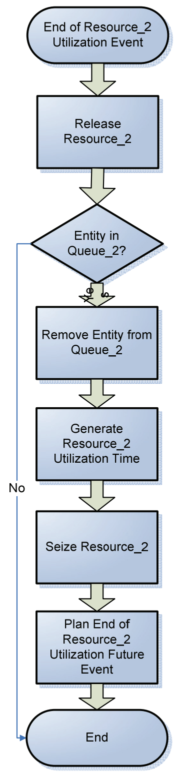

Figure 3 - End of Resource_2 Utilization Event Flowchart

The software tool would recognize these flowcharts upon the following (simplified) dialog:

```
Add Event
Event Name? Entity Arrival
Is it an Arrival Process? Y
Time Between Arrival? Exponential (3)
It will follow a Resource Utilization? Y
Which Resource? Resource_1
Probability associated? 100%

Add Event
Event name? End of Resource_1 Utilization
Is it an Arrival Process? N
Resource_1 Utilization Time? Random (5,9)
Number of Resource_1 Units Available? 3
It will follow a Resource Utilization? Y
Which resource? Resource_2
Probability associated? 80%

Add Event
Event name? End of Resource_2 Utilization
Is it an Arrival Process? N
Resource_2 Utilization Time? Normal (4, 0.25)
Number of Resource_2 Units Available? 2
It will follow a Resource Utilization? N
```

Following this dialog and according to the previous reference to the text file creation by the software tool, the text file below would be created:

```
Entity Arrival expo(3) 0 [Resource_1] [100]
Resource_1 Random (5,9) 3 [Resource_2] [80]
Resource_2 Normal (4,0.25) 2 [] []
```

As stated above and upon fully recognition of the characteristics of the text file constructed, the user could directly identify the text file corresponding to the flowcharts developed (skipping the dialog phase) and the software tool would automatically generate the Java Basic Simulation Facility program. This text file would also act as a starting point to future modifications of previous used examples.

This tool has been largely used in simulation classes, with informatics, information systems and industrial engineering students. They are asked to construct flowcharts for various examples and to use our tool. After a few initial enhancements, the tool has been generating correct Java programs. This has been the true verification procedure for this simulation tool.

4. CONCLUSION

The software tool presented shows three particularly interesting features:

- It is based on simple flowcharts that follow Event Scheduling Simulation philosophy
- It uses simple what-if questions to implement the respective flowcharts
- It automatically generates a Java computer program to perform the mimic of the system under analysis

These features, together, would contribute to

- the generalization and a better understanding of the use of simulation
- the automatic generation of simulation programs (*Java programs*)

The generalization and better understanding of the use of simulation would be accomplished once our tool only requires expertise in a basic simulation approach – Event Scheduling, thus incorporating simple flowcharts defining the system and its functioning rules.

Then, these simple flowcharts are translated into the software tool through a very simple questionnaire that is responsible for the automatic generation of a Java computer program that performs the mimic of the system and evaluates corresponding efficiency measures.

As far as future work is concerned, the authors would include in this tool the facility to deal with multiple queues examples. This feature would then recognize the ability for a resource to use entities from multiple queues, implementing both possibilities:

- when a resource uses entities from multiple queues
- when a resource uses an entity selected from a set of queues

REFERENCES

Clementson, A.T., 1982. *Extend Control and Simulation Language - ECSL*, Users Manual. Birmingham: CLE, COM Ltd.

Dias, L.S. and Rodrigues, A.G., 2002. Towards Simplicity in Modeling for Simulation. *Operational Research Society, Simulation Study Group, Two days Workshop*. March 20-21, Birmingham, UK.

Dias, L.S., Rodrigues, A.G. and Pereira, G.B., 2005. An Activity Oriented Visual Modeling Language with Automatic Translation to Different Paradigms. *Proceedings of the 19th European Conference On Modelling And Simulation (ECMS 2005)*, pp. 452-461. June 1-4, Riga (Latvia).

Dias, L.S., Pereira, G.B. and Rodrigues, A.G., 2007. A Shortlist of the Most Popular Discrete Simulation Tools. *Simulation News Europe*, 17 (1), 33-36.

Gilbreth F.B. and Gilbreth L.M., 1921. Process Charts - First Steps in Finding the One Best Way to do Work. *Presented at the Annual Meeting of The American Society of Mechanical Engineers*, New York, USA.

Nassi and Shneiderman, 1973. Flowchart techniques for structured programming, *ACM SIGPLAN* Notices, 12.

Pidd, M., 1992. *Computer Simulation in Management Science*. 3rd Ed. John Wiley & Sons, Inc.

Thesen, A., 1978. *Computer Methods in Operations Research*. New York: Academic Press.

Thesen, A. and Travis, L.E. 1990. Introduction to simulation (tutorial session). *Proceedings of the 22nd Conference on Winter Simulation*, pp. 14-21. December 09 – 12, New Orleans (Louisiana, United States,). O. Balci, Ed. IEEE Press, Piscataway, NJ.

Tocher, K.D., 1963. *The Art of Simulation*, UNIBOOKS – English Universities Press.

AUTHORS BIOGRAPHY

Luís S Dias was born in 1970 in Vila Nova de Foz Côa, Portugal. He graduated in Computer Science and Systems Engineering in the University of Minho, Portugal. He holds a PhD degree in Production and Systems Engineering from the University of Minho, Portugal. His main research interests are Operational Research, Simulation and Systems Visual Modeling.

Guilherme A B Pereira was born in 1961 in Porto, Portugal. He graduated in Industrial Engineering and Management in the University of Minho, Portugal. He holds an MSc degree in Operational Research and a PhD degree in Manufacturing and Mechanical Engineering from the University of Birmingham, UK. His main research interests are Operational Research and Simulation.

José A Oliveira was born 1966 in Matosinhos, Portugal. He studied Mechanical Engineering at the University of Porto, Portugal. He graduated with a Ph.D. in Production and Systems Engineering at University of Minho, Portugal. His main research interests are Optimization with Heuristic Methods in Systems Engineering.