Chapter 5
Specification and Description Language for Discrete Simulation

Pau Fonseca i Casas
Universitat Politècnica de Catalunya - Barcelona Tech, Spain

ABSTRACT

Designing a new simulation model usually involves the participation of personnel with different knowledge of the system and with diverse formations. These personnel often use different languages, making more difficult the task to define the existing relations between the key model elements. These relations represent the hypotheses that constrain the model and the global behavior of the system, and this information must be obtained from the system experts. A formalism can be a powerful tool to understand the model complexity and helps in the communication between the different actors that participate in the definition of the model. In this chapter we review the use of the "Specification and Description Language," a standard and graphical language that simplifies the model understanding thanks to its modular nature. To do this we present a complete example, representing a simple queuing model that helps the reader to understand the structure and the nature of the language.

INTRODUCTION

The principal motivation of Operations Research (OR) is to understand the behavior of systems through representative models. Alternatives can be evaluated without interacting with or disturbing reality to choose the most appropriate system modification. Models can be used not only to compare alternatives but also to predict the

DOI: 10.4018/978-1-4666-4369-7.ch005

behavior of a known system when variables are modified within a specific range. Hence, operations research simulation models can yield solutions for possible future situations. The simulation field contains a variety of different paradigms to model a real system; this study uses a specification and description language to represent discrete-event simulation models. Based on (Law & Kelton, 2000) Law & Kelton (2000), Guasch, Piera, Casanovas, & Figueras (2002)(Guasch, Piera, Casanovas, & Figueras, 2002), and (Fishman, 2001) Fishman (2001), discrete-event simulation consists of three major methodologies: process interaction, activity scanning, and event scheduling.

The event-scheduling methodology is based on an initial description of events. Events are the elements of the model that cause modifications in the state variables of the model. A function is defined for each event. The functions are executed when the time associated with an event is the same as or very close to that shown on the model clock1. The events are sorted in an event list by their time and priority. The time between two events is irregular. The simulation clock therefore jumps from one time to the next without following any set pattern (Law & Kelton, 2000).

To model a system that is composed of a server that receives elements over time according to a specific distribution, different events can be generated that represent the elements entering the system if the intervals between the events are known. Similarly, if the distribution that defines the time that the server requires to process an element is known, an event that defines the service time for a specific element can be generated. This information is illustrated in Table 1.

From Table 1, the diagram (time chart) shown in Figure 1 can be generated.

This figure shows an M|M|1, following Kendall's notation (Kendall, 1953), is a representation of a system composed by one machine proceeded by a queue (FIFO)2. The initial state of the server is free, and the events shown in Table 1 define the behavior of the model. In the event-scheduling paradigm, one event is processed in each simulation loop and the procedure related to this event is executed. The pro-

Table 1. Events

Time between Arrivals		**Service Time**	
a1	35	b1	40
a2	12	b2	30
a3	29	b3	30
a4	47	b4	20
a5	12	b5	30

Related Content