History of Simulation

Evon M. O. Abu-Taieh
The Arab Academy for Banking and Financial Sciences, Jordan

Asim Abdel Rahman El Sheikh
The Arab Academy for Banking and Financial Sciences, Jordan

Jeihan M. O. Abu-Tayeh
Ministry of Planning, Jordan

Hussam Al Abdallat
The Arab Academy for Banking and Financial Sciences, Jordan

INTRODUCTION

The great philosopher Aristotle said, "If you would understand anything, observe its beginning and its development." Therefore, understanding simulation requires observing its history.

Accordingly, simulation can be understood in many ways: "Simulation is the use of a model to represent over time essential characteristics of a system under study" (El Sheikh, 1987). Another definition is "Simulation is the imitation of the operation of a real-world process or system over time" (Banks, 1999).

Simulation was known long before computers. According to Araten et al. (1992), "The first econometrics model of the United States economy was constructed by J. Tinbergen in 1939." Later, as computers developed in the late 1950s and early 1960s, a spawn of computer simulation methodologies and approaches came to life. Computer simulation, like any industry, both affected and was affected by the development of different programming languages and computer capabilities and advances.

This article will first give a background about simulation in general, then it will discuss the classical simulation methodologies. We will address the current trends in simulation by presenting currently used Java-based simulation languages. In this regard, the classical simulation methodologies discussed in this article include the three-phase approach, activity scan, process interaction, event scheduling, transaction flow approach, Petri nets, and Monte Carlo. The languages discussed are simjava, DEVSJAVA, JSIM, JavaSim (J-Sim), JavaGPSS, Silk, WSE (Web-enabled simulation environment), SLX, and SRML (simulation reference markup language). As such, this article will tackle the history of the approaches and methodologies while shedding light on the genealogy of the simulation languages.

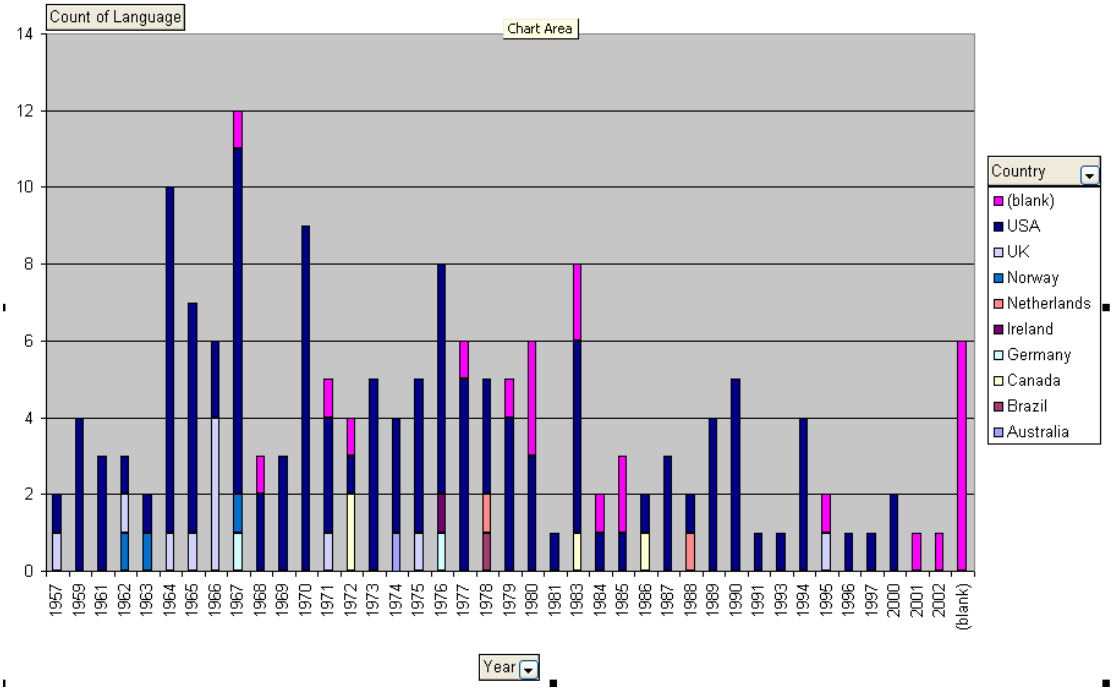
BACKGROUND

Defining simulation in its broadest aspect, we can say that it embodies a certain model to represent the behavior of a system, whether that may be an economic or an engineering one, with which conducting experiments is attainable. When studying models currently used, such a technique enables management and allows us to take appropriate measures and make fitting decisions that would further complement today's growth sustainability efforts, apart from cost decreases as well as service delivery assurance. As such, the computer simulation technique contributed to cost decline, depicted cause and effect, pinpointed task-oriented needs and service delivery assurance, explored possible alternatives, identified problems, proposed streamlined measurable and deliverable solutions, provided the platform for change-strategy introduction, introduced potential prudent investment opportunities, and finally provided a safety net for conducting training courses. Yet, simulation development is hindered due to many reasons. Like a rose, the computer simulation technique does not exist without thorns. Simulation reflects real-life problems; hence, it addresses numerous scenarios with a handful of variables. Not only is it costly and liable to human judgment, but also, the results are complicated and can be misinterpreted.

Within this context, there are four characteristics that distinguish simulation from any other software-intensive work according to Page and Nance (1997). First, simulation uses time as an index variable. Second, one of the objectives in simulation is to achieve correctness. Third, simulation software is involves computational intensiveness. Last but not least, the use of simulation is not typical; in fact, "no typical use for simulation can be described" (Page & Nance, 1997, p. 91).

Hence, there are many methodologies and approaches that simulation practitioners use when working on a simula-

Figure 1. Simulation languages in chronological order



tion project: the three-phase approach, activity scan, process interaction, event scheduling, transaction-flow approach, Petri nets, and Monte Carlo (Abu-Taieh & El Sheikh, 2007). Based on the previously mentioned methodologies and approaches, as well as other programming languages, more than 170 simulation programming languages and more than 60 simulation packages (Abu-Taieh & El Sheikh) were developed, for example, GPSS, GSP, GASP, SIMULA, and so forth. The simulation programming languages are reflected in *Figure 1* in terms of the date and country of origin for each programming language.

CLASSICAL SIMULATION APPROACHES AND METHODOLOGIES

There are many simulation approaches and methodologies. For this purpose, the most familiar will be thoroughly discussed, namely, the three-phase approach, activity scan, process interaction, event scheduling, transaction flow approach, Petri nets, and Monte Carlo.

Three-Phase Approach

The first simulation modeling structure is known as the three-phase method. It was described by Keith Douglas Tocker in 1963 in his book *The Art of Simulation* (Odhabi, Paul, & Macredie, 1998), and then discussed in detail by Pidd and Cassel (1998).

Tocker introduced the three-phase approach through the General Simulation Program (GSP), which is considered as the first language effort (Nance, 1995; Pidd, 1998). In 1966, Tocker introduced wheel charts, which were later replaced by the activity cycle diagram and the language CSL (control and simulation language), which represented a simpler approach called activity scan (Nance). Tocker claimed that the structure would enable automatic programming simulation as Nance (1995) states was obvious in the development of OPS-1, OPS-2, OPS-3, and OPS-4, which are simulation languages intended for non-computer-programmers developed by students in MIT.

The three-phase approach has, as the name suggests, three phases—the A phase, B phase, and C phase—as seen in Figure 2. Each phase will be further discussed next.