A Brief History of Simulation Revisited

David Goldsman (Georgia Tech)
Richard E. Nance (Virginia Tech)
James R. Wilson (NC State)

www.ise.ncsu.edu/jwilson/simhist10.pdf

December 7, 2010
Acknowledgment

- This presentation is jointly sponsored by the WSC Foundation, www.wscfoundation.org, and the Simulation Archive, tinyurl.com/simarchive.

- The WSC Foundation was established by the WSC Board of Directors in 2003 as an independent organization charged with developing and managing a Fund to help ensure the continuance and high quality of the Winter Simulation Conferences.

- With support from ACM-SIGSIM, ASA, INFORMS-SIM, the Dept. of Information Systems and Computing of Brunel Univ., the ISE Dept. of NC State Univ., and many individuals, the Simulation Archive is a publicly accessible collection of books, papers, and other source material documenting the development of discrete-event computer simulation since the 1940s.
Outline

1. Precomputer Era: From Buffon to World War II (1777–1945)


4. Summation
Our objective is to highlight people, places, and events that have marked the development of discrete-event and Monte Carlo simulation.

We also seek to motivate others to document their historical contributions in places like the Simulation Archive.
Disclaimer: Caveat Auditor

- This is a “work in progress,” and the associated *Proceedings* article is at best a placeholder.

- Our coverage of the history of simulation is highly selective and subjective, with emphasis given to a few key individuals and events.
Georges-Louis Leclerc, Comte de Buffon (1707–1788), was a celebrated French naturalist who anticipated many of the ideas of Darwin and Lamarck on evolution.
Buffon is remembered in the history of probability theory for his famous *needle problem*—the first example of a simulation experiment; see

The Buffon Needle Problem

**Buffon’s Needle Problem:** If a floor has equally spaced parallel lines a distance $d$ apart and if a needle of length $\ell$ is tossed at random on the floor where $\ell \leq d$, then what is the probability that the needle will intersect a line?
The Buffon Needle Problem (Cont’d)

If $U$ is the distance from the needle’s midpoint to the nearest line and $A$ is the angle between the needle and the line defining $U$, then Buffon interpreted a “random” toss to mean that

$$
\begin{align*}
U &\sim \text{Uniform}(0, d/2), \\
A &\sim \text{Uniform}(-\pi/2, \pi/2), \\
U \text{ and } A &\text{ are independent;}
\end{align*}
$$

(1)

and he correctly calculated $\Pr\{\text{Needle intersects a line}\}$ as

$$
\int_{-\pi/2}^{\pi/2} \int_{0}^{d/2} I\{(\ell/2) \cdot \cos(a) \geq u\} \frac{2}{\pi d} \, du \, da = \frac{2\ell}{\pi d}.
$$

(2)

If $\ell = d/2$, then with a large number of needle tosses the fraction of successes can be used to estimate $1/\pi$. 

Buffon’s needle-tossing experiment is the earliest example of using independent replications of a simulation to approximate an important physical constant—a technique revived by Ulam in 1946 to design the hydrogen bomb.

Buffon also proposed a version of the needle problem for a rectangular grid of equally spaced parallel lines, but his derivation is incorrect. Laplace gave a correct solution to what is now known as the **Buffon-Laplace needle problem**; see

William Sealy Gosset (1876–1937), trained in mathematics and chemistry, was a brewer with Arthur Guiness, Son & Co. Ltd. and made numerous contributions to statistical methodology in his spare time.
Gosset was faced with the problem of maintaining consistent quality of Guinness’s ale and stout based on data with the following drawbacks:

- small sample sizes; and
- measurements that are not independent.

Thus he was working in quality control 25 years before the Shewhart chart.

Gosset arranged to spend 1906 studying under Karl Pearson at University College London, but he quickly discovered that Pearson’s large-sample statistical methods were inadequate for Guinness’s problems.
Gosset’s Approach to Small-Sample Process Control

To estimate the mean $\mu$ of a normal population based on a random sample $\{X_i : i = 1, \ldots, n\}$ with sample size $n$ in the range $4 \leq n \leq 10$, he proceeded as follows:

a. He calculated the sample mean and variance,

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \text{and} \quad S^2 = \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{X})^2. \quad (3)$$

b. He derived the mean, variance, skewness, and kurtosis of $S^2$, and he showed that these characteristics of $S^2$ exactly match those of a Pearson type III curve; then he guessed that $S^2$ has this distribution when the $\{X_i\}$ are normal.
c. He showed that if the \( \{X_i\} \) are sampled from a symmetric distribution, then

- the statistics \( \overline{X} \) and \( S \) are uncorrelated; and
- the statistics \( \overline{X}^2 \) and \( S^2 \) are also uncorrelated.

d. Since the normal distribution is symmetric about its mean, Gosset guessed that in random samples from a normal distribution, \( \overline{X} \) and \( S^2 \) must be independent.
Gosset’s Approach to Small-Sample Process Control (Cont’d)

Gosset used his results a–d to show that the probability density function of the ratio

\[ Z = \frac{\bar{X} - \mu}{S} \]  \hspace{1cm} (4)

based on a random sample of size \( n \) from a normal distribution with mean \( \mu \) has the form

\[ f(z) \propto \frac{1}{(1 + z^2)^{n/2}} ; \]  \hspace{1cm} (5)

and from (5), he computed tables of selected percentile points of the distribution of \( Z \) for sample sizes in the range \( 4 \leq n \leq 10 \).
Gosset’s Approach to Small-Sample Process Control (Cont’d)

To validate his results, Gosset conducted a precomputer simulation experiment by randomly sampling from a population of left middle finger lengths of 3,000 habitual British criminals obtained from New Scotland Yard.

- These measurements were written on 3,000 pieces of cardboard, thoroughly shuffled, and drawn at random to yield a randomly ordered list of the entire population.

- Each consecutive set of 4 measurements from this list was taken as a sample of size \( n = 4 \), so that there were 750 such samples.
For each sample of size 4, Equations (3)–(4) were used to compute the corresponding $Z$ statistic; and a histogram of the resulting 750 $Z$-values was superimposed on the density (5) with $n = 4$ as shown below.

Guinness allowed Gosset to publish his results, provided he used a pseudonym and no proprietary data was used. These results were published under the pseudonym “Student” in 1908.
Postscript on the Discovery of Student’s $t$-Distribution

- R. A. Fisher finally published a mathematically rigorous derivation of Student’s $t$-Distribution in 1921.

- This inaugural application of simulation to industrial process control is a remarkable example of the synergy of simulation-based experimentation and analytic techniques in the discovery of the exact solution of what is arguably a classical industrial-engineering problem.
Stanisław Ulam (1909–1984) was a Polish mathematician who worked on the Manhattan Project and originated the Teller-Ulam design of thermonuclear weapons.
Ulam’s Epiphany

Ulam was fond of card games, and in 1946 he was led to consider a simulation approach for estimating the probability that a Canfield solitaire laid out with 52 cards will come out successfully.

In 1946 Ulam also performed detailed calculations showing that Edward Teller’s initial design for the hydrogen bomb was inadequate.

With the availability in 1946 of ENIAC, one of the first electronic computers, Ulam realized that computer-based simulation could be used effectively to estimate the intractable mathematical integrals arising in the design of a workable hydrogen bomb.
Using Monte Carlo Methods to Design the Hydrogen Bomb

Ulam proposed an alternative design for the hydrogen bomb that he and Teller ultimately perfected and patented.

Ulam convinced John von Neumann and Nicholas Metropolis to work with him on developing the “Monte Carlo” (simulation) methods required to implement the Teller-Ulam design for thermonuclear weapons.
Postscript on the Term “Monte Carlo”

Metropolis coined the term “Monte Carlo,” explaining that

\[\ldots\] Stan had an uncle who would borrow money from relatives because he “just had to go to Monte Carlo.” The name seems to have endured.

Apparently the only archival documentation on the genesis of this term is the following:

Transition in the Computer Era

- Hardware focus predominates (1946–1960)
  - ENIAC; Mark I, II; Univac I; Whirlwind; ILLIAC; ...
  - “Coders” use binary instructions, then octal (machine language)

- Programming begins to emerge in mid-1950s
  - Assembly language ("automatic programming")
  - FORTRAN: a scientific language
  - COBOL: a business language
  - ALGOL: a “universal” language

- Underlying mathematics and statistics recognition

- Simulation analysis—DG

- Simulation “programming” (modeling)—REN
The Central Problems of Digital Simulation

As simulation started to develop as a modeling tool, it became time to sketch out a new research agenda for the growing field. Richard W. Conway of Cornell University provided the first widely recognized general framework for doing so.
Based on their extensive research and practical experience involving manufacturing simulations in the 1950s and early 1960s, R. W. Conway, B. M. Johnson, and W. L. Maxwell of Cornell University laid out the central problems of digital simulation in two seminal papers:


Conway, Johnson, and Maxwell said that computer simulation problems fall into two broad categories—the *construction* of the simulation model, and the *analysis* of the simulation results.
The main problems in using simulation include the **strategic** problem of designing a simulation experiment and the following **tactical** problems on how to run the simulations specified in the experimental design:

- **a.** Determining when a simulation is in equilibrium (steady state) so that any transients caused by the simulation’s initial condition have died out;

- **b.** Estimating the precision (variance) of simulation-based estimators of steady-state performance; and

- **c.** Performing precise comparisons of alternative system simulations.
For the start-up problem a, Conway (1963) proposed the first widely used rule for truncating (deleting) simulation-generated observations that are contaminated by initialization bias.

For the variance-estimation problem b, Conway (1963) proposed the method of batch means, which is still widely used in practice and is the basis for much ongoing research.

For the comparison problem c, Conway (1963) rejected ANOVA and proposed the use of statistical ranking-and-selection procedures, which are now widely used in practice and are the basis for much ongoing research.
Conway’s work directly spurred a tremendous amount of research in the field.

The subsequent analytical and methodological work of George S. Fishman and A. Alan B. Pritsker owes some of its inspiration to Conway.

In fact, as Barry Nelson states in his article “Stochastic Simulation Research in Management Science,”

...the foundation for the field—not just the work that has been published in Management Science—was provided by two papers published long before simulation had its own department in the journal... the seminal papers of Conway, Johnson, and Maxwell (1959) and Conway (1963).

Barry then goes on to trace their impact through eight award-winning papers that appeared much later in Management Science.
The Central Problems of Digital Simulation (Cont’d)

Let’s look at some of those papers . . .

- Lavenberg and Welch (1981), “A perspective on the use of control variates to increase the efficiency of Monte Carlo simulations” [c]
- Meketon and Heidelberger (1982), “A renewal theoretic approach to bias reduction in regenerative simulations” [c]
- Shahabuddin (1994), “Importance sampling for simulation of highly reliable Markovian systems” [c]
And some more-recent worked directly descended from Conway…

Big Question: How to generate proper random variates quickly and efficiently?

- Fishman’s text: Succinct categorization of a number of useful techniques
- Devroye’s text: Major compendium of all techniques to date
- Schmeiser’s work: Fundamental contributions to development of algorithms for univariate and multivariate random variates; easy-to-read survey
  - Poisson, gamma, etc.
  - M/M/1 waiting-time process

- Beautiful tricks
  - Ahrens–Dieter normal generator
  - Various nonparametric generators
Contributions to Output Analysis

Big Question: How to analyze resulting output of complicated systems?

The problem is. . . Simulation output is almost never independent, identically distributed, or normal.

- Schmeiser formalizes the properties of the batch means method
- Fishman formulates a time-series approach to output analysis
- Fishman and Iglehart formulate the regenerative method
- Schmeiser formulates the method of overlapping batch means
- Schruben formulates the method of standardized time series
- Wilson and Pritsker deal with initialization bias
- Several authors work on ranking-and-selection and optimization problems
Simulation Modeling: Application Driven

- **Manufacturing**
  - United Steel Companies, Ltd: [K.D. Tocher](#) (General Simulation Program and Activity Cycle Diagrams);
  - General Electric Manufacturing Simulator (GEMS)
  - Hughes Aircraft: IBM Job Shop Scheduler (JSS)
  - United States Steel: [Philip Kiviat](#) (General Activity Simulation Program (GASP))

- **Telecommunications**
  - Bell Telephone/IBM: [Geoffrey Gordon](#) (GPSS)

- **Military**
  - RAND Corporation: [Harry Markowitz](#), Philip Kiviat
K. D. Tocher: A Simulation Pioneer

- U.K. Ministry of Aircraft Production (1942–1945)
- National Physical Laboratories (1945–1948)
- Imperial College, London (1948–1957)
- United Steel Companies, Ltd. (1957–1980)
The General Simulation Program (GSP)

- Originating in 1958, GSP consisted of a set of routines recognized by Tocher ("Toch") as necessary in all simulation programs.
  - Initialization
  - Time and state advance
  - Report Generation

- Time and state control was a major issue
  - Machines cycled through states of "busy," "idle,"...
  - State progression could be time-based or state-based

- Key reference:
Tocher’s Three-Phase Activity-Scanning Method for Timing Control

The state of each machine evolves over time in three phases: (A) advancing time to the next scheduled event that is “bound to occur” and that may change the machine’s state (this is called a “B-event”); (B) processing the associated B-event; and (C) processing “conditional” events (called “C-events”) that are not scheduled for specific times but are instead subject to prespecified conditions on machine state.
Tocher’s Three-Phase Activity-Scanning Method for Timing Control

- The B-events are strictly based on *time*, and the C-events, on *state*.

- Repeated scans of the C-events occur until time must be advanced for the next event to occur.

- The characterization of the time and state interaction to produce events is the crucial requirement of all modeling languages used in discrete-event simulation.

- Failure to recognize the important distinctions in terminology used in defining time and state interactions plagued the discrete-event simulation for over twenty years; see

Wheel Charts and Activity Diagrams

- Tocher sought a simple model representation to capture the cycle of state progressions.
  - Focus on the major expense items: equipment
  - Represent state transitions that are costly, e.g. “busy” to “idle” or “available” to “unavailable”
  - The Wheel Chart of Tocher, 1966 offers a simple, instructional modeling tool that evolves

- Activity Cycle Diagrams
  - Adapted by academics and practitioners in UK
  - Assume a major conceptual role for several languages
Activity Cycle Diagram of Steelmaking Process
Activity Cycle Diagram of Steelmaking Process
Manufacturing Systems: An Application Driver

- In his 1958 Ph.D. dissertation at UCLA, Alan Rowe conceived of using simulation to investigate scheduling rules in the job-shop environment. His conceptions were tested in an industrial setting, but generality in application proved a difficult objective.

- Drawing on Rowe’s experience, Harry Markowitz and Mort Allen developed the General Electric Manufacturing Simulator (GEMS). Lessons learned by Markowitz were helpful in his later developments of SIMSCRIPT (described later).

- Donald G. Malcolm chaired two symposia on the potential impact of simulation on industrial engineering; see

During the period 1960–1962, John Colley, Harold Steinhoff, and others developed a model of Hughes Aircraft’s El Segundo fabrication plant based on the IBM Job Shop Scheduler (JSS).

This simulation was used to test dispatching rules using operational data, and the results were used in “near-real-time” mode to guide production decisions in the ensuing shift; see Bulkin, M. H., J. L. Colley, and H. W. Steinhoff, Jr. 1966. Load forecasting, priority sequencing, and simulation in a job shop control system. *Management Science* 13 (1): B29–B51.
Manufacturing Systems: Kiviat and GASP

- Created by Philip Kiviat at U.S. Steel in 1961, originally coded in ALGOL, was based on FORTRAN II.

- Kiviat’s graduate work at Cornell was an influence.

- Elements (people, machines, orders) comprised the model.

- Intended to bridge gap between engineers and programmers, GASP used a graphical interface.

- Unique feature was provision of a regression equation for possible input value generation.

- GASP II (1963) appeared with Pritsker (1967).
Geoffrey Gordon joined the Advanced Systems Development Division of IBM in 1960 as Manager of Simulation Development; and during the period 1960–1961, he introduced the General Purpose System Simulator, which was later renamed the General Purpose Simulation System (GPSS).

GPSS was designed to facilitate rapid simulation modeling of complex teleprocessing systems involving, for example, urban traffic control, telephone call interception and switching, airline reservation processing, and steel-mill operations.

GPSS is notable for its effective use of specialized block diagrams for graphically representing the flow of entities through the system.
GPSS exploits a *process-interaction approach* to simulation, whereby we model the sequence of activities in which temporary entities (transactions) engage permanent entities using resources in moving through the system or waiting for the release of resources because of competition (interaction) with other entities for the resources required to complete the production process.

GPSS was a macro language implemented as an interpreter, which slowed execution time but permitted rapid changes.

GPSS/H, the compiled version by Jim Henriksen, became the predominant implementation in the mid-1970s.

Model description is intended to follow a paradigm of “program what you see using the macros provided”
The Enduring Legacy of Geoffrey Gordon and GPSS

- Because of its remarkable ease of use and the marketing efforts of IBM, GPSS was distinguished as the most popular simulation language of its time—and the process-interaction approach to simulation is still the method of choice for many large-scale simulations of complex industrial operations.

- The 1967 forerunner of the Winter Simulation Conference (WSC) was the Conference on Applications of Simulation Using the General Purpose Simulation System (GPSS), which in subsequent years was expanded to include papers on any simulation language or any aspect of simulation applications.

- WSC is now the premier international forum for disseminating recent advances in the field of system simulation.
Military Systems: RAND and SIMSCRIPT

- The RAND Corporation developed SIMSCRIPT a general language with simulation capabilities.

- Harry Markowitz, using his experience with GEMS is considered the principal designer.

- User interface consists of three forms: definition, initialization, report generation.

- Model description in terms of entities with attributes that are members of sets.

- SIMSCRIPT I produced FORTRAN statements, a 1.5 version compiled into assembly code.

- SIMSCRIPT II redesigned with Philip Kiviat.
SIMSCRIPT II: Markowitz and Kiviat

- Harry Markowitz (later a Nobel laureate in Economics) was principal designer early; B. Hausner provided programming expertise.
- Philip Kiviat became principal designer in 1965–1966; R. Villanueva became principal programmer.
- Proprietary versions of the RAND product:
  - SIMSCRIPT II Plus; SIMSCRIPT II.5
Ole-Johan Dahl and Kristen Nygaard worked for the Norwegian Defense Research Establishment as operations research analysts from the late 1940s to the early 1960s, when they both moved to the Norwegian Computer Center.

During the period 1961–1967, Dahl and Nygaard coinvented object-oriented programming through their development of the general-purpose programming languages SIMULA I and SIMULA 67, which include special features designed to facilitate the description, interaction, suspension, and reactivation of processes.

SIMULA is an extension of ALGOL that was funded heavily by Univac.
Dahl and Nygaard’s Development of SIMULA

- Based on a process description, SIMULA expanded the process interaction conceptual framework to permit added flexibility in “class” declaration.

- SIMULA has been not only one of the most influential simulation languages but also the programming language with the most pronounced effect on the development of software engineering.

- The introduction of SIMULA led to a fundamental change in the techniques for designing and programming software systems, resulting in applications code that is reliable, scalable, and reusable.
SPLs were the primary modeling tool, but a user interface did not always force programming skills.

John Crookes identified over 132 SPLs (?)

More models were developed in GPLs than SPLs.

Communication among SPL developers was quite good; among SPL users, very limited.

- Inversion of theory and application (Kiviat)
- Inhibition for research in simulation theory.

Conceptual modeling frameworks not treated.
Core Challenges in Model Construction

- Early limitations have been largely eliminated
  - RAM management through reductions in cost
  - File management provided by Simulation Programming Languages (SPLs)

- Other concerns remain but seem less important
  - Control of error from forced discretization of data

- Persistent problems
  - Model sustainment (reliance on GPL; SPL mismatch)
  - Model and program documentation

- Growing realization of need for “modelware” support and importance of Conceptual Frameworks
Conceptual Frameworks and Modeling

- CFs provide support for transference of system characteristics to model representation.
  - Linkage between time and state relationship must be captured.
  - Object, state, and time are the requisite components for descriptive attention.
  - Objectives and assumptions exert a major influence on the utility of one CF versus another.

- Fit of the CF with an application domain (including objectives and assumptions) can complicate task.
  - Ability to effect the transference from system and objectives to model representation can be jeopardized.
  - Visualization of “world view” supported in another CF can be quite difficult.

- Challenge of estimation of model complexity or project effort.
**Example System Problem**

**The Machine Interference Problem**

A set of semi-automatic machines that fail randomly are assigned to a single operator, who repairs them according to different strategies (or policies). The number of machines that should be assigned is affected by machine downtime cost, operator idle time cost.

**Repair Strategies:**

1. Repair in order of failure
2. Repair nearest failed machine
3. Patrol machine perimeter to make repair

**Assumptions:**

1. Travel time is negligible
2. Travel time is significant
3. Exponential inter-failure time
4. Exponential repair time
Example System Problem

Objective is to minimize Total Cost (TC)

\[
\min TC = \text{operator idle cost} + \text{machine downtime cost}
\]

Consider combinations of repair policies and assumptions

<table>
<thead>
<tr>
<th>Policy</th>
<th>Assumption</th>
<th>Resulting Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repair in order of failure (FFFS)</td>
<td>Travel time is 0</td>
<td>M/M/1 analytical</td>
</tr>
<tr>
<td></td>
<td>Travel time &gt; 0</td>
<td>G/M/1 analytical</td>
</tr>
<tr>
<td>Repair closest machine (NDFS)</td>
<td>Travel time &gt; 0</td>
<td>G/M/1 analytical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(complexity suggests simulation)</td>
</tr>
<tr>
<td>Patrolling repairman</td>
<td>Travel time &gt; 0</td>
<td>G/M/1 analytical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(complexity strongly suggests simulation)</td>
</tr>
</tbody>
</table>
Expansions and Extensions of SPLs (1970–1979)

- Capability for combined continuous/discrete event
  - GASP IV (1973) – Pritsker and Hurst
  - C-SIMSCRIPT (1976) – Delfosse (CACI)

- Non-IBM Versions of GPSS
  - GPSS/NORDEN, NGPSS – Reitman
  - GPSS/H (1977) – Henriksen

- PL/I Packages – SIMPL/I (1972), SIML/I (1979) - MacDougall

- Conceptual extensions
  - Transaction-processing user interface (SIMSCRIPT II.5)
  - SLAM (1979) – Pritsker and Pegden

- Activity-cycle Diagram to Program Generators
  - CAPS/ECSL (Computer Aided Program Specification) – Clementson
  - DRAFT (Interactive Program Generation) – Mathewson
    - FORTRAN
    - SIMULA
    - SIMSCRIPT

- Systems-theoretic modeling formalisms – Zeigler
  - Characterization of model components
  - Relationships among models

- Interest in Model Development Environments
  - Graphical output reporting of results
  - Techniques of model verification and validation – Sargent
Transition in Focus: Program $\rightarrow$ Model

- **Conceptual Framework Focus**
  - Event Scheduling: Event Graph Modeling – Schruben (1982)

- **Application Derived Representation**
  - Niche products: circuit design, scheduling, etc.

- **Influential Simulation Books**
Some general observations on the evolution of simulation since the eighteenth century—

- The significant advances in the theory and practice of system simulation over the past three centuries have been driven by researchers and practitioners working in a wide diversity of disciplines.

- Much of this work has been motivated by practical applications, and it has necessarily been interdisciplinary.

- The future vitality of the field depends on the preservation and extension of this heritage.