Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

Keynote Address: A concise history of the ups and downs of simulation

Julian Reitman
University of Connecticut, Stamford Campus
and
George Mason University, Fairfax Virginia

ABSTRACT

The history of the Winter Simulation Conference provides insight into the working environment and intellectual and managerial attitudes during the formative period of simulation software development. It also suggests a basis for comparison with current practices. This history provides a springboard from which to extrapolate a few predictions for simulation capabilities of the future.

BACKGROUND

A personal note is one way to start reviewing history which one has witnessed and participated in: the evolution of discrete event simulation. The intention is to present a historical summary. It is hoped that this excursion into the history of the Winter Simulation Conference will provide insights into our formative period, a basis for a comparison with current practices, and the opportunity to make a few predictions for the near term. One characteristic of the personal view is that the biases of the individual highlight some experiences over others. I hope that those omitted do not feel slighted and I apologize for my shortcomings.

The background that I brought to simulation was that of engineering: the design of electronic hardware and the use of analog computers with their close man-machine interaction. The engineering was very pragmatic; for example, one area of experience involved the design and installation of a complex, for those days, test system. When the production line stopped because the test system failed, you were very quickly aware of the situation. It was not the usual background for a digital computer system designer, but in the mid 1950s there was no usual background.

In the 1950s, computers were beginning to be used for census data reduction, defense systems, accounting, and scientific calculation. At that time I was designing computer based Airline Reservation Systems, real-time systems with computers interfacing and interacting with communication systems. One of our problems was simple to formulate. The queuing theories that were well developed and employed for the design of telephone systems did not produce results which matched our user's service requirements. Queuing theory was based on the expectation that the interarrival time between transactions and the holding time for these transactions would approximate a negative exponential. This would be mathematically convenient. Unfortunately, our field data for peak periods did not conform to these characteristics, and peak periods are the weak link in real-time systems. One technique we tried was to attempt to model the system with discrete event computer simulation.

With the tools that were available, our approach was to use an IBM 650, assembly language, and a team of mathematician, systems engineer and programmer. Results: we accomplished less than half of what we set out to do, took twice as long and overspent our budget at least twice. Conclusion: computer simulation was not a useful tool in the 1950s. Simulation took too long to get results, needed too many skilled people, and as a result cost a considerable amount in both personnel and computer time. And most disheartening, the results were ambiguous. There was no incentive to continue the simulation experiment. Simulation was for others, not for those with practical problems and deadlines.

After joining Norden, I was assigned once more to the system design of a real-time computer system. This was a system for the FAA to distribute weather information to general aviation. Again the question was how to meet peak requirements for a system that would not conform to current queuing theory. We visited a number of potential computer hardware suppliers. IBM suggested they knew the answers since they had simulated this class of problem. In October 1961 IBM presented the "Gordon Simulator" to us at a closed meeting, and in December 1961 Geoffrey Gordon presented his paper at the Fall Joint Computer Conference on a General Purpose Systems Simulator (GPSS) [1,2]. We were willing to try this new tool, but our expectations were quite minimal.

In those days, 1962, software was provided by IBM with the hardware. And so, in due time, a one week training course was provided for us to learn GPSS-I. It became my responsibility to select a problem, get the card deck from IBM, and set up a trial. Our group of three engineers worked on the problem and a programmer loaded the GPSS-I card deck and ran the example problem. With very little difficulty, the example problem ran. Next we started to model a problem that did not lend itself to a mathematical solution. Suddenly we were in business. We were able to construct the model, simulate our problem, and obtain answers. The entire process was accomplished in only six weeks. A new tool had become available for system designers.

Now that a tool was available to aid in system design, I set out to use it. The attitude at Norden was supportive and soon I was involved in the simulation of the repair of military systems using GPSS. What was even more significant was that as models began to be produced for outside groups, a dedicated simulation activity was established. We were concentrating our energies on building models and then running them to understand the systems. We felt quite alone since the users of GPSS from IBM were concentrating on aspects of computer systems very different from our systems.

Geoffrey Gordon's concept was that the actual designers would use GPSS. They would learn the language and use it as needed. Our experience, after extensive experimentation in training design engineers, was to the contrary. Only a few engineers were willing to learn and become competent in a simulation language or for that matter to write a FORTRAN simulation program. Engineers preferred to communicate their problem to programmers or a simulation group. Early simulation groups were established at: Boeing, Martin Marietta, Air Force Logistics Command, General Dynamics, Hughes Aircraft, Raytheon, Celanese, Exxon, Southern Railway, and the computer manufacturers - IBM, Control Data, National Cash Register, and UNIVAC.

A CONFERENCE IS NEEDED

One significant need was to communicate with other simulation groups. IBM facilitated communication by telling us who the others were and roughly what they were doing. But the real beginnings of interaction among the GPSS simulation groups occurred though the IBM users' group conference, SHARE. At these conferences we were able to exchange information with other simulation groups. SHARE, by that time, was already a huge meeting, so those interested in simulation had only one session. It was hard to justify going to a distant three or four day conference for one session.

Meanwhile, at Rand Corporation Harry Markowitz, Bernard Hausner, and Herbert Karr produced a version of SIMSCRIPT in 1962 [3]. The Air Force was particularly interested in SIMSCRIPT to simulate their inventory problems. Elsewhere, there were other approaches. In England J. Buxton and J. Laski developed CSL, the Control and Simulation Language [4], early versions of SIMULA were developed in Norway by O. Dahl and K. Nygaard [5,6], and Don Knuth and J. McNeley produced SOL - A Symbolic Language for General Purpose System Simulation [7]. Ken Tocker provided us with a short book on the ART OF SIMULATION [8].

The characteristics of this period were quantities of simulation language developments and few efforts to coordinate and compare the different approaches. There was, also, no organized activity to help users get started or provide guidance. The first step to address these limitations was to look at simulation languages. This was done at a Workshop on Simulation Languages at Stanford University in March of 1964. Then at the International Federation for Information Processing (IFIP) Congress in New York in May of 1965 there was a discussion of languages and applications which in turn led to another Workshop at the University of Pennsylvania in March of 1966. One result of this Workshop was the realization that a narrower conference on the uses of simulation was needed.

A SERIES OF CONFERENCES BEGINS

In response to these needs, an organizing group was established composed of members of SHARE, Joint User's Group of ACM, and the Computer and Systems Science and Cybernetics Groups of IEEE. This group organized the November 1967 Conference on Applications of Simulation using the General Purpose Simulation System (GPSS). Some items from that antecedent of the WSC:

> The first day had a Keynote address and three sessions with 12 papers;

The second day had ten parallel sessions, a panel, and a session on Man-machine interfaces for GPSS;
There were two luncheons, with Geoffrey Gordon speaking on "The Growth of GPSS" at one of them;
Registration was $30 and included the two luncheons, and this was at the New York Hilton;

Hotel rates were $14-25 for a single;
The intention was to limit attendance to two hundred and twenty-five attendees, but four hundred and one showed up.

Obviously, there was a need for future conferences and for a greater number of papers to be presented. Of equal importance was the need to publish a conference record. Encouraged by success, the organizing group set out to make the conference format broader, include other languages, and provide a conference digest. The 1968 Second Conference on the Applications of Simulation was held in December of 1968 in New York at the Hotel Roosevelt with over seven hundred attendees. For that conference, what is today known as SCS became a sponsor and a 368 page Conference Digest was published. That Conference became the first one to address, in great variety, the many aspects of discrete event simulation. There were a total of seventy-eight papers presented at twenty-two sessions. A conference format had emerged that would last.

Some of the memorable items were:

Keynote Address of "Why is Top Management Difficult to Convince?"
Sessions with papers on Statistical Considerations, random number generation for GPSS/360, languages - SIMSCRIPT II, SIMULA 67, SPURT, a simulation tutorial, and the Case for FORTRAN - A Minority Viewpoint;
Sessions covered transportation, computer systems, manufacturing applications, reliability and maintainability, graphics and GPSS modifications, simulation and human behavior, distribution systems, communications, urban systems, gaming models, job shops, materials handling, marketing models, languages for modeling computer systems, facility planning models, and simulation and ecology.

The 1969 Third Conference on the Applications of Simulation was held in December in Los Angeles. One sign of becoming established was that both AIIE and TIMS joined as sponsors. Among the new items were GASP and a session on health systems. The 1970 and 1971 Fourth and Fifth Conferences were again, and for the last time in New York. The notable aspect of the Fourth Conference was the first GPSS tutorial by Tom Schriber. The Fifth Conference became the first to be titled the WINTER SIMULATION CONFERENCE. The number of tutorials grew with Alan Pritsker covering GASP II and Yen Chao SIMSCRIPT. Also, an education session was added since many schools were offering courses in both continuous and discrete event simulation.

A radical departure was instituted for the next two conferences, they were held in January. It had been thought that travel budgets were exhausted by the end of the year, so instead of December, meet in January. There was no significant difference. After the San Francisco and Washington meetings, the conference returned to December in 1976. Both the 1976 and 1977 conferences were located at the National Bureau of Standards near Washington, D.C. NBS joined the sponsors and provided the facilities. 1976 was also significant for the first SIMSCRIPT tutorial by Ed Russell. The range of topics grew with increased emphasis on the analysis of simulation results and in 1977 sessions on agricultural and military systems were added. There was also an increased interest in the internal workings of the languages. One example was AN IMPROVED EVENTS LIST ALGORITHM presented by Jim Henriksen.

WSC BECOMES AN INSTITUTION

The growth in the number of sessions which had started in 1967 with twelve was an indication of progress. The number of sessions had doubled by 1971 and continued to rise to about forty sessions in 1977. In 1983 there were sixty sessions and there it has stayed.

A sign of the growing maturity in the field was a Panel Discussion at Miami in 1978 on the FAILURES OF SIMULATION, focussing on what can and does go wrong and a paper on MANAGING SIMULATION PROJECTS. 1979 brought the conference back to the West Coast, San Diego, and East in 1980 to Orlando. A pattern was emerging. There were many more tutorials, and papers were organized into tracts of sessions for beginners, intermediate, and advanced practitioners. The Conference was established, attracting an audience, and gaining momentum. Successful simulations were being reported at the Conferences. Simulation was becoming part of the changing American pattern of careful study and financial analysis before a commitment to a new facility. Models were increasingly used to design new plants and to plan the flow of work in these new facilities. The influence of graphics became more marked and a number of vendors used the conference exhibit space to demonstrate the advantages of their system by actually bringing a working computer to the conference site. Technology had moved so far that simulation, for those who were skilled in the art, became quicker, cheaper, and much more responsive to the designs of the model constructor.

Since 1969, the Conference Proceedings have provided a written record of the papers presented. The size of these Proceedings has varied from 326 to 1051 pages. Since 1984, there has been a single volume in hard cover. Previously, there were two volumes, soft covered. Under the IEEE open order policy, the Proceedings have been purchased by approximately two hundred libraries.

NOW - THE HISTORY LESSON

In 1987, after twenty-six years with Norden, I accepted early retirement and began a new career as a systems consultant and teacher of the history of technology. So, with my new history point of view, let us review the story of simulation. Would GPSS, SIMSCRIPT, and SIMULA evolve today as they did almost thirty years ago? Of course not, conditions change. Remember my early experience when we spent twice as much, achieved half what we intended and took twice as long, and used only company funds. That is a useful starting point.

Today, just the proposal to management to study the possibility of simulation would consume our entire exploratory budget in constant dollars. Of course if we got funded, we would be more adequately endowed; but the probabilities are low. But, assume the project is funded, now, if we are late in meeting a milestone, we have to stop work while we explain the delay and reprogram the effort in still greater detail. Would our results be different. Would we have developed a useful simulation tool? The answer is no. There is no way a group can start from scratch and write a complete simulation system and have it become productive with a reasonable expenditure of funds and a limited time frame. Today, we could use higher level languages, but imagine using COBOL, ADA, or FORTRAN. Those are the standard languages and they might be mandated. It is very difficult to imagine having both adequate time and funds.

The next episode in our saga is IBM making GPSS available. What were the 1961 ground rules? IBM supplied the card deck; we had to load it and try the sample model. No monies were exchanged. In those days there were overhead accounts to cover such trivia. Today there had better be a valid charge number. There might be a chance to try a program product before purchase. The actual construction of a model and the testing of it might still be absorbed by the project. It was not a significant amount, and we did much of it in the evening, as we would now.

The FAA did not have the funds to implement the study; that is still highly probable. We had learned something about simulation, but where to apply it. Management supported a part-time effort for three years until 1965 when there was a funded simulation effort. It is very unlikely that something as strange and different as simulation would be supported for three years in today's R and D climate.

SIMULATION LANGUAGES

A more fundamental question is would a GPSS, SIMSCRIPT, or SIMULA be developed? Geoffrey Gordon was at the IBM's Advanced System Development Laboratories. He did not have to justify the return on investment for a program product. He did not have to meet standards for documentation, testing, and training. There was considerable internal interest in the system. So, GPSS might be developed today by IBM. It just would not be released to the public. At least not until IBM was sure that it would not provide their competition with a useful tool. Also, with only internal users, the product would have been unreliable for a longer time and may have had fewer users sticking with it till most of the bugs were out.

SIMSCRIPT was a very different case. It was developed at RAND and then released to the world with a disclaimer that there were bugs still in it and they were being worked on. Today the Department of Defense would consider the program SECRET and restrict the dissemination. The result would be very few users and the bugs would be waiting for someone to find them. Result, no one would use such an unreliable product, assuming that they could get it, or even know that it existed.

SIMULA was different. The Norwegians developed it with government funds and the concept of distributing it to the world. The problem in the United States was a lack of familiarity with ALGOL. It is hard to visualize our using ALGOL, but under today's rules it might have been mandated. It is conceivable, following this rationale, that today our only widely distributed simulation language would be SIMULA.

However, the picture changed when software became a program product. The modern era began. What has been reviewed are the developments before that dramatic discontinuity. The conferences have provided an audience for a very large number of simulation packages.

SIMULATION ENVIRONMENT

One difference between then and now that is hard to evaluate is the relative contributions by industry and academe. In the early days, the development of languages was a product of industry. What the academics were interested in were the statistical questions. How long should the simulation continue? When have the initial conditions been passed? How reliable are the simulation experiments? There was one other topic which turned out to interest some of both groups. How to evaluate, program, and test pseudo-random number generators. Cooperation provided both a stronger foundation and confidence in simulation.

There were some significant positive advantages that we had in those early days. The customer knew very little and believed just about anything that was presented. If the computer printed it out, especially in the form of lengthy tables, it must be true. No need for fancy graphics, just a few bar charts, although there were some extensive efforts to use the IBM 2250 display unit with GPSS. It was too expensive to be used by the great majority.

Also, the problems being simulated were simpler. Model sizes were smaller. In fact, just asking the questions needed to construct the model would frequently define the problem sufficiently to halt any further efforts. The process of defining the problem in terms suitable for constructing a model provided a degree of discipline that otherwise would have been lacking.

The capabilities of the computers had a significant effect on the productivity of the modelers. Initially, we used decks of tabulator cards. All systems were batch mode. A card deck was submitted along with the control cards to have the program tape loaded. Then you waited until the printout came back. With GPSS, one frequent error resulted in generating too many transactions with an error dump of all the transactions. The quantity of paper that we consumed was significant. Unfortunately, the queue discipline for printout was based on the size of the printout. The result of loading the tape, running the model, and getting an error was, frequently, one run a day. Debugging a model took quite some time if your development style was to let the computer find the errors. Desk checks were the alternative.

WHAT'S NEXT

Our world is influenced by both changes in hardware and software. Absorbing the rapid changes that are still going on in the hardware will take some time. When current hardware capabilities are fully used, we will be able to:

> Use a network of processors, PC or microprocessor elements, to retain the parallelism of the real world.
>
> Increase the number of processors to speed up the simulation to meet realistic real-world speed and capacity requirements.
>
> Include in the processing network some processors using actual real-time data along with those doing the simulation. This will allow some elements to be performing discrete event and others continuous simulations.
>
> Dedicate graphical co-processors to enable the viewer to observe the simulation in the simulation time reference.

Simulation language capabilities will have additional features to reduce the effort to build the model and to allow greater flexibility when running the simulation. These features when combined with output services will allow:

> Less effort for model development as there will be generic elements that have multi-functions and are provided with computer aided instructions for their use and suitable for aggregation into a model or family of models.
>
> Increased model capability and variety through the use of previously constructed and debugged generic sub-elements ready to be recompiled and aggregated into a variety of different and complex model combinations.
>
> Relatively unsophisticated users to be trained by computer instruction associated with the generic model elements to define and comprehend the capabilities and limitations of the aggregated model.
>
> A full variety of graphical presentations, including three dimensional views, hidden lines, zoom in and out, and color.
>
> The quantity of processors needed to simulate the problem to be independent of the user's input and dependent on the needed resources taken from the resource pool.
>
> Greater man-machine interaction when running the model to permit selecting and observing a number of entities during the simulation and with adjustable levels of detail or granularity.
>
> A complex real-time model to be run and observed in real-time and at specific choke points to continue the real-time processing and advance to fast-time for man-machine or AI alternatives to predict the possibility of clearing the choked elements while still optimizing the entire system's performance.

CONCLUSION

The Winter Simulation Conference has provided a window on the development of a significant use of computers. We have seen the languages improve, the applications become more varied, and the results applied. The conference has provided a forum for the exchange of ideas and a continuing presence. It is rare in today's world to see so many sponsors combine to support one conference. I have been fortunate to be part of that process and to have made so many friends while participating in the progress of simulation.

REFERENCES

1. Gordon, Geoffrey (1961). A General Purpose Systems Simulator. In proc. EJCC, Washington D.C., pp. 87-104. New York: Macmillan.

2. Wexelblat, Richard L. ed. (1981). History of Programming Languages: GPSS Session pp. 403-37. New York: Academic Press.

3. Markowitz, Harry M., Bernard Hausner, and Herbert Karr (1963). SIMSCRIPT - A Simulation Programming Language. Englewood Cliffs, New Jersey: Prentice-Hall.

4. Buxton, J. N., and J. G. Laski (1962). Control and Simulation Language. Computer Journal 5(5).

5. Dahl, O.-J., and K. Nygaard (1966). SIMULA - an ALGOL - based Simulation Language. CACM 9(9): 671-8.

6. Wexelblat, Richard L. ed. (1981). History of Programming Languages: SIMULA Session pp. 439-93. New York: Academic Press.

7. Knuth, Don E., and J. L. McNeley (1964). SOL - A Symbolic Language for General Purpose System Simulation. IEEE Transactions on Electronic Computers ED-13(4): 401-8.

8. Tocker, Kenneth D. (1963). The Art of Simulation. London: English Universities Press.

Julian Reitman teaches the history of Science/Technology at the University of Connecticut, Stamford campus and at George Mason University in Fairfax, Virginia and has taught Computer Simulation at University of Bridgeport and New York University. He received a BEE from City College of New York in 1949 and a MEE from New York University in 1954.

At Norden Systems, 1961-1987, he introduced GPSS and improved graphics, data bases, and contributed to:

- Computer and communications message switching systems
- Transportation systems: Urban Personal Rapid Transit (PRT),
- Intercity surface mass transit, and air traffic control
- Systems effectiveness of airborne and shipboard systems
- Performance prediction for surface-to-air missile systems
- Analysis of radar system built-in test adequacy
- Traffic capacity of centers processing Income Tax returns
- Yield prediction for integrated circuit production
- Design of computer systems for real-time military applications
- Seminars in England, Germany, Netherlands, Israel, Italy, and Peoples Republic of China.

He is the author of Computer Simulation Applications, 1971, and has written chapters and articles for books, IEEE Spectrum, Computer, Simulation, and presented numerous papers. He helped start WSC as Program Chairman of the 1967 Conference and General Chairman in 1968, and continued on the Board of Directors until 1985. Other activities have included: editorial board of the IEEE Press, associate editor of IEEE Transactions on Systems, Man, and Cybernetics (SMC), and president of IEEE SMC Society. He is a Senior Member of IEEE, SCS, AAAS, and the Society for the History of Technology. From 1943-45, he served in the U. S. Army Air Force.