TEACHING SIMULATION TO TEN THOUSAND STUDENTS
AMERICAN-EUROPEAN COOPERATION AND PERSPECTIVES

Ingolf Ståhl   Richard G. Born

Stockholm School of Economics   OM&IS Department
Box 6501   Northern Illinois University
SE- 11383 Stockholm, SWEDEN   DeKalb, IL 60115, U.S.A.

Henry Herper

Institute for Simulation and Graphics
Otto von Guericke University
Magdeburg, 39102, GERMANY

ABSTRACT

This paper deals with the cooperation between three European and American simulation teachers, who together have taught simulation to over ten thousand students in five countries. They have, based on student feedback, developed an educational version of GPSS, the General Purpose Simulation System. This simplified system, aGPSS, has proved to be very easy to learn and also to use, for example, in student projects. The three teachers have also together written simulation textbooks in English and German

1   INTRODUCTION

This paper deals with the cooperation between three European and American simulation teachers, Ingolf Ståhl, Sweden, Richard Born, USA, and Henry Herper, Germany, below called S, B and H respectively. After first teaching simulation on their own for several years, they entered in 1991 into cooperation for developing an easy-to-learn system for simulation, later to be called aGPSS, used in several countries. Together they have taught simulation to more than ten thousand students. Counting the years from when they started teaching simulation to today and adding up this time, they can together look back on roughly a hundred years of teaching simulation.

To this number of students and years one could have added the many thousands of students and the jointly almost hundred years of experience from teaching simulation of Thomas Schriber, University of Michigan, USA, and of Peter Lorenz, University of Magdeburg, Germany, who have both been greatly instrumental for the cooperation project discussed in this paper. Their textbooks and teaching of GPSS and of using Proof Animation in connection with GPSS have, as will be discussed further below, had a great influence on us, the three authors of this paper. They have, however, in contrast to us three, in their teaching, usually to more advanced students than ours, remained with "traditional" GPSS and have not moved to aGPSS, the focus of this paper. We want, however, to start this paper by acknowledging our deep gratitude to these two persons.

The cooperation, eventually leading to the aGPSS system and the textbooks on this, started at the Winter Simulation Conference in Phoenix, 1991. We shall hence below first discuss our simulation teaching activities prior to this conference, next our meeting at this conference, then continue with discussing what happened after this and end with some general perspectives.

2   TEACHING BEFORE THE WSC'91

Ståhl (S) started his teaching of simulation in the mid-80s when teaching the main course in Management Science at SSE (the Stockholm School of Economics). This was a full-semester course, i.e. corresponding to 3 US credits and requiring roughly four full weeks of study. The textbook used was MacMillan and Gonzales (1973), containing a chapter of 77 pages on GPSS, written by Schriber. To learn more about GPSS, S acquired Schriber (1974), the "Red Book", and became quite fascinated by this. This book has a total of 27 case studies, which all refer to practical problems, but yet appear easy to understand.

S wanted the students to use GPSS for making their own small models. At this time, GPSS, then GPSS/360, was available only on IBM mainframes, which SSE lacked, so the students had to walk a mile to the computer center at another college. The students enjoyed GPSS, but did not like this walk. Hence, when SSE acquired a PR1ME minicomputer, which lacked a GPSS system, S decided to make his own "mini-GPSS" system, which started as a small subset of GPSS/360. The system was programmed in FORTRAN, the best programming language at this time on the PR1ME. It was a proper subset of GPSS/360, with two exceptions. It avoided a major critique against IBM GPSS, namely that strange results are obtained if a GENERATE block is immediately followed by a SEIZE block. There are separate internal waiting lists for **each** server instead of one single current event list, leading to higher efficiency.

In 1983-85, S was visiting professor at Hofstra University, NY, where he taught an elective full-semester course in simulation five times. This course was devoted almost completely to GPSS, which was first run on an IBM computer using GPSS/360. S had, however, already earlier started to transfer his mini-GPSS from the PR1ME to the new IBM PC micro-computer. The new system, called micro-GPSS, could be used in 1984.

Since the whole course was devoted to simulation, S found it important to have the students make a simulation project out in a company, with a focus on modeling, data collection etc. As S expanded this subset of GPSS, he found it important to include features that the students needed in their projects and yet keep GPSS easy to learn. The primary guide on what to include was the 27 case studies in Schriber (1974). S could rewrite not only all of the 27 case studies, but also about 99 percent of all programs in other GPSS textbooks, with roughly the same amount of code.

When S was back in Sweden, GPSS was soon introduced as a quarter of a full-semester course taken by all 300 students at the SSE, many with little computer background. Here S used several teaching assistants. S got a great amount of feedback on what was difficult to learn and also what was difficult to teach. Based on this feedback, S started to simplify the syntax in several respects. Before 1985, micro-GPSS was a pure subset of GPSS/360, with the two exceptions mentioned above, but now it became a streamlined version.

For users of early versions of GPSS, it might be of interest to note the following: An IF block with "straight logic" replaced the TEST block, with e.g. IF Q$Joe=4,BYE instead of TEST NE Q$Joe,4, BYE of IBM GPSS. Further, WAITIF lock=U was used instead of GATE NU lock. With students tending to forget commas, GOTO BYE and GOTO BYE,0.15 were used instead of TRANSFER ,BYE and TRANSFER .15,,BYE. In order to simplify and make the concepts stronger, a single LET block replaced the ASSIGN, SAVEVALUE, PRIORITY and SELECT blocks. The collection of statistics was simplified by SEIZE Joe,Q replacing QUEUE JoeQ, SEIZE Joe, DEPART JoeQ. Furthermore, features that lead the students to make difficult-to-find logical errors were eliminated. With students reporting unclear errors, S developed a system for error reporting with 500+ error messages.

Micro-GPSS was in the late 1980s taught at several universities and colleges in Sweden, among others in logistics at Chalmers University of Technology. In 1990 a text book (Ståhl 1990), based on micro-GPSS, was published by Prentice Hall and students in many countries then ordered the software.

In the summer of 1991 S attended the first course ever in Europe of Proof Animation of Wolverine Software, then still as a beta version. S realized the potential of combining GPSS with animation and changed micro-GPSS so that one with a number of simple HELP blocks could produce animation trace files for Proof. A number of very simple animation examples were produced.

When **Born** (B) taught at Kaneland High School (Maple Park, IL) from 1967 through 1981, he began to teach computer programming in 1972, first by taking punched cards from his students to Northern Illinois University (NIU) for processing, bringing the printed output back to the students the next day. Then in 1975, Kaneland leased a General Automation computer, having FORTRAN IV with WATFIV. Students could then punch their programs onto IBM cards and then enter them via a card reader, and get their printed output within a few minutes. B had the students write some very rudimentary FORTRAN programs doing simulation modeling that he had read about in the Encyclopedia of Computer Science. This was part of a six-semester sequence of courses in the Kaneland Computer Science Curriculum that B developed for the "Gifted Program" for students with exceptional academic ability at Kaneland.

After getting his Ph. D. in 1988, B. started teaching at NIU. Here B first taught simulation modeling in his telecommunication classes in 1989, using CACI COMNET II.5 and LANNET II.5 software. CACI still exists with SimScript and SimProcess, but the COMNET and LANNET software are no longer sold or supported. Some of these simulations were run in an NIU computer lab with about 24 PCs running a variety of scenarios non-stop 24 hours a day. The teaching was closely connected with B's research on telecommunication networks, leading to several articles, *inter alia* Born and Kenevan (1988) and Born (1991), which deal with simulation modeling of telecommunication networks and how B used the CACI Products Company's LANNET II.5 simulation tool in his telecommunications courses. Born and Kenevan (1988) contains the listing of four GPSS models, each dealing with a specific type of very-large-scale, regular, static interconnection network. B's teaching of GPSS for network simulation was largely influenced by his Ph. D. thesis (Born 1988).

As regards more general purpose simulation, B first began teaching a course in simulation modeling at NIU in 1990. The course centered on the textbook by Watson and Blackstone (1989), with problems assigned to the students from this textbook. This book has a large chapter on GPSS, but also chapters on SLAM and GEMS. In his class, B used mainly GPSS, but also compared GPSS to these other languages.

Herper (H) started teaching simulation in 1988 at what was then the Technical University of Magdeburg, at that time in the GDR. The first courses, of around 48 class room hours of lectures and practice sessions, were with students of computer science, in Germany called Informatics. After some time the course was also taught to students of Mechanical Engineering and to students of Business and Economics. The main software used initially in the course was a GPSS version called SIMDIS, running on IBM/360 similar mainframes called ESER. SIMDIS had been developed since the 1970's, first in Dresden and later in Magdeburg. SIMDIS had certain extensions to IBM GPSS/360 and V, mainly access to data bases and additional commands for matrix manipulation. H used Frank and Lorenz (1979) as the textbook for his course. Starting in 1990, a PC based version of GPSS, programmed in Pascal, called SIMPC, was also used. This was based on the development by one of H's colleagues, Thomas Schulze. SIMPC was slightly smaller than SIMDIS, lacking e.g. data bases, but had other advantages, such as alpha-mosaic graphics. Starting in 1991, H also used Witness when teaching simulation. Witness had the advantage of animation. It was used in special courses having mainly students of Production Engineering.

3   START OF COOPERATION

The cooperation between S, B and H started at the Winter Simulation Conference in Phoenix in December 1991. S, who had met Tom Schriber some years earlier, was advised by him to attend the conference, but being too late to submit a paper, S decided to rent space as a vendor. The cost of the space turned out to be just a small part of the total cost, since S also had to rent a table, a computer, power outlets, drapes, chairs, etc. and yet S's booth was by far the simplest of all booths. At the booth S and his (future) wife demonstrated simple animation models produced by micro-GPSS and Proof. They also sold a diskette with the micro-GPSS system together with a very simple booklet (Ståhl 1991) for $10.00. One would not believe that such a venture was meaningful, but it truly was, since one of the few buyers was Rich Born (B) and this was the start of over 20 years of cooperation. At the same conference there were two young German simulation scientists, Henry Herper (H) and his colleague Volkmar Hinz. With the German reuni-

fication just one year old, they were for the first time in the West. With S speaking German and all three interested in GPSS, there was a natural meeting of minds. S was then invited to come to Magdeburg for the first Simulation Fachtagung in March of 1992. S here presented a paper on animation with micro-GPSS and Proof (Ståhl 1992).

Here cooperation continued and S came back to Magdeburg in 1993 for the foundation meeting of the GPSS-Users Group Europe. Here not only H and S, but also Hinz, Lorenz, Schulze and Schriber presented papers, all relating to GPSS (ASIM 1993). S continued coming to the Magdeburg conferences most years in the 1990s and cooperation with H developed further, in particular in two directions. S had continued to develop his booklet on micro-GPSS further to fit the class of 300 students at SSE. At the SSE it was regarded suitable that interested students could also get some of their education in other languages than Swedish and English, and hence SSE offered the GPSS module of 10 class room hours in German and a score of students signed up. H then came for a week during each of five years to teach this GPSS module. In connection with this, H then started to translate S's booklet into German.

H could use this booklet also for his teaching of micro-GPSS in Magdeburg. At first H taught GPSS/H together with Proof to the students of Informatics. In the state of Sachsen-Anhalt, Informatics was included in the high school curriculum. The last year of this curriculum allowed the student a choice of an area of special studies (for around 30 – 40 class room hours), and simulation was one of these optional areas in which H in 1994 started to teach future high school teachers. Since these future teachers had much less background in programming than the regular computer science students, GPSS/H was found to be less suitable and H instead gave courses in micro-GPSS.

Parallel to this cooperation between H and S, cooperation between B and S got under way. After attending WSC'91, B made the decision to start using micro-GPSS in his classes for business students. A key reason was that for his business students, micro-GPSS would be easier to learn than other GPSS versions. For the first years B used Ståhl (1991), but this booklet was soon revised and expanded, with feedback from B, gradually into Ståhl (1996). B also started to produce a set of slides to accompany this teaching of micro-GPSS. They were initially developed using software available by IBM for the earliest of the Windows operating systems. These slides in turn inspired S into further improvements of his booklet. In addition to micro-GPSS, B included Promodel into his simulation class. Promodel, with animation integrated, allowed for much easier use of animation than micro-GPSS with Proof. At the end of the course, after students had learnt both micro-GPSS and Promodel, the students returned to a business problem, solved in the first lecture by hand, to solve it using both micro-GPSS and Promodel (Born 2003).

B also taught simulation using spreadsheets. B wrote a working paper (Born 1997) on the use of spreadsheets to perform inventory analysis, which B used extensively in his simulation classes in the late 1990s and early 2000s. It shows among other things the complexity of simulation modeling using spreadsheets. In his teaching at SSE, Ståhl also tried to use spreadsheets for simulation of the Monte Carlo type. The limitations of simulation using spreadsheets in a dynamic context are discussed in Born and Ståhl (2013). The main problem is to represent stochastic time delays, e.g. in deliveries or payments. In GPSS this is done e.g. by a simple block ADVANCE A,B, which can allow for a very great number of possible time delays. As exemplified by the spreadsheet model in Born (1997), only three possible delay times are included in order to keep the model from becoming too complicated.

4 ADDITION OF A GUI - WEBGPSS AND WINGPSS

Up to 1998, micro-GPSS was, just like e.g. GPSS/H, a purely text-based system. Both B and H transmitted feedback from students that it was very desirable that input could also be made using a **G**raphical **U**sers **I**nterface, where the student from a menu of symbols could choose the building blocks of the program. Then in 1997, a new Swedish Foundation for Computers in Education was looking for university developed software that could be placed on the Web. This was a chance to get financing of roughly $150,000 to have a GUI based GPSS put on the Web. In 1999 the first tentative versions of such a GUI-based system, WebGPSS, based on micro-GPSS, was presented, with the client developed in Java. The programs

were first developed and translated into text code on the client PC. The code was then run on a remote server using the micro-GPSS system. Parallel to this, H developed another GUI system, WinGPSS, in Delphi in Magdeburg, and running in Windows on student PCs, but using the same micro-GPSS engine.

 In both systems, blocks were chosen by clicking on a symbol in a menu, leading to a block diagram. This point-and-click method allows for a faster model build-up than the drag-and-drop method. By next clicking on a block in the block diagram, a dialog with syntax explanations is opened to allow for the input of the operands of the block. The execution of the code is in both systems carried out by the micro-GPSS "engine", which produces output files in ASCII format, which WebGPSS and WinGPSS then turn into tables, histograms and graphs. The output, shown under several tabs, is more readable and understandable than the output generated by standard GPSS. It is easy to print and save each result tab.

 WebGPSS was made generally available on the Web, run on a Swedish server, and was used in a few Swedish colleges, but also in several high schools, in particular for individual projects in the senior high school year. WinGPSS was used in the German state of Sachsen-Anhalt, where high school students in their senior year chose between optional courses from different areas of computer science, one of which is *Introduction to Modeling and Simulation*. The students in this course carried out simulation projects, using WinGPSS and Proof Animation (Ståhl and Herper 1999). The first version of WebGPSS of 1999 only allowed for 16 of the 22 block types of Micro-GPSS, but enough to handle the features in Ståhl 1996. The first version of WinGPSS allowed for all 22 blocks, but was more restricted e.g., as regards the control statements. The symbol menu of WinGPSS is seen to the very left in Figure 1, while the symbol menu of WebGPSS of 1999 is seen to the right of this. A similar figure is commented in more detail in Ståhl et al. (2011), a paper which deals with the general history of GPSS and includes information also on other GPSS systems like IBM GPSS, GPSS/H, GPSS World and SLX, the advanced successor of GPSS.
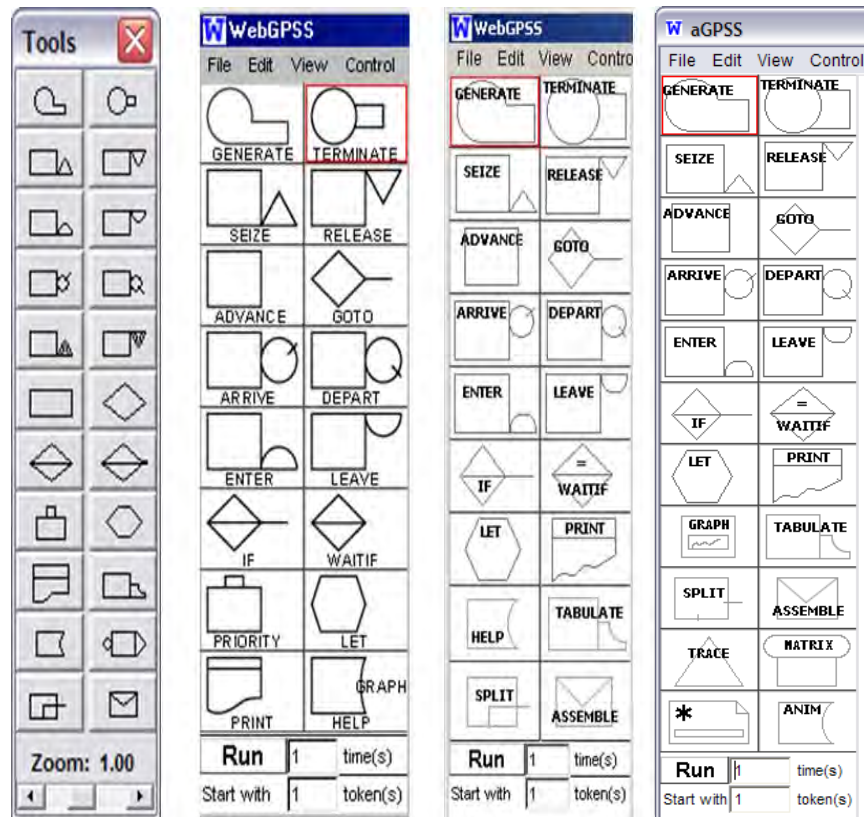

Figure 1: Symbol Menus in WinGPSS, the Two Versions of WebGPSS/and of aGPSS

Since 1999 there have also been many improvements of the WebGPSS system. In 2003, we extended the system so that it would have the same functionality as provided by the 22 blocks in micro-GPSS. In Figure 1 we see the WebGPSS symbol menu of 2003 with a total of 18 blocks to the right of the original 1999 menu with 16 blocks. There are here three new blocks, TABULATE, SPLIT and ASSEMBLE. The reason that there are only 18 blocks (instead of 22) is first of all that the PRIORITY block is not needed, since a stronger LET block can set, increase or decrease priority. The functionality of the SELECT block is now also handled by the LET block. LET p$best=MIN,Q,1,6 conveys the idea that p$best is the shortest of six queues better than SELECT MIN 1,1,6,,Q of traditional GPSS. Parameters are now given symbolic names and the PREEMPT and RETURN blocks are now part of the SEIZE and RELEASE blocks.

Finally, in 2011 there was one more change in the symbol menu, as shown by the menu to the very right in Figure 1. Since the word HELP proved to be confusing to the students, it was eliminated. The HELP block was earlier used for graphs, tracing and matrix handling. We have now given GRAPH, TRACE and MATRIX their own symbols. We finally have, at the bottom of the right column, a block, ANIM, for the interface between Proof Animation and what we now, as discussed below, call aGPSS.

The introduction of WinGPSS and WebGPSS had consequences also on the teach ware. The micro-GPSS booklet was now changed to *Simulation Made Simple with WebGPSS* (Ståhl 2003), with information about the GUI. This book was, like the earlier micro-GPSS books, without pictures. In the German translation of the earliest chapters made by H, there were, however, many figures of the block diagrams produced by WinGPSS (Ståhl and Herper 1998; Born *et. al.* 2005). Furthermore, B started in 2003 to produce new slides, now in PowerPoint. Inspired by H's translations and by B's slides, S included many pictures in a new book in 2004, now in direct cooperation with B (Born and Ståhl 2004). The first booklet with only the first lessons was also presented as a paper at the WSC'04 (Born and Ståhl 2004b). This booklet was then gradually expanded to include 30 lessons (Born and Ståhl 2007).

5   WINGPSS + WEBGPSS = AGPSS

The most recent development in the cooperation process is the move towards merging WinGPSS with WebGPSS. WinGPSS has two strong advantages over WebGPSS. It has a dialog for animation with Proof and it allows for block based animation, i.e. it allows the user to follow the movement of the transactions through the system. A disadvantage is that WinGPSS, not developed further since 2004, lacks many control statements. WebGPSS has as regards control statements facilities for experiments and optimization, for warm up and for antithetic random numbers. WebGPSS allows for running the model many times with an automatic calculation of the limits within which the universal average will lie with, for example, 95% probability. It also allows for a limited amount of optimization with a graphic representation of these confidence intervals for each of many alternatives. For the case of a comparison of two alternatives, it will determine if one alternative is better than the other one with 97.5% probability.

Another recently introduced improvement in WebGPSS is the possibility to place blocks in different columns. While other simulation systems mainly use a drag-and-drop buildup of the block diagram, WebGPSS uses, as mentioned, a point-and-click buildup. The system originally placed a block just below the last inserted block, except for GENERATE blocks, always placed at the top. The disadvantage was that all blocks in a segment were placed in **one** single column. The new system now allows the user to move the blocks to any desired position by the arrow keys. The information about the **relative** position of a block that is not placed just below the earlier block is in the computer code indicated by an extra part of the comment. This has the advantage that all information about the placement of the blocks is in ASCII code, allowing for very compact code, yet containing all information necessary for the block diagram.

Another important reason for merging WebGPSS with WinGPSS is that WebGPSS since 2004 is run in a way that is very similar to that of WinGPSS, i.e. as a stand-alone version under Windows. The code produced by WebGPSS was, as mentioned above, first run on a remote server in Sweden, by use of the micro-GPSS engine, When many students used it at the same time, serious problems arose, in particular when students ran the same program many times to establish confidence levels. With students in many far

away countries, there were also transmission problems. We hence decided that we had to have WebGPSS just running directly under Windows. The server module is then also run on the user's computer. With the micro-GPSS engine programmed in FORTRAN, and not in Java, to allow for fast execution, these disadvantages of being on the web outweighed in this case the advantages of the web. By thus moving to Windows, WebGPSS has come closer to WinGPSS. Because of this, we have regarded the name **aGPSS** as more suitable than the old name WebGPSS. The **a** indicates that is a first system for beginners.

In order that aGPSS should also have the main advantages of WinGPSS, we have first of all allowed for a dialog for the interface to Proof Animation. We have, as mentioned earlier, included a block ANIM in the symbol menu. Clicking on this block, the dialog to the right in Figure 2 is obtained. This can be compared with the corresponding dialog of WinGPSS to the left. It should be mentioned that Write in WinGPSS is equally well handled by the PRINT block and tracing by the TRACE block in aGPSS.

Figure 2: Animation Dialogs in WinGPSS and aGPSS

In order to have a block animation system similar to that of WinGPSS, shown for an M/M/1 system to the left in Figure 3 below, we allow for a completely automatic buildup in aGPSS of Post-Run trace and layout files for Proof Animation from **any** aGPSS program, without an ANIM block.
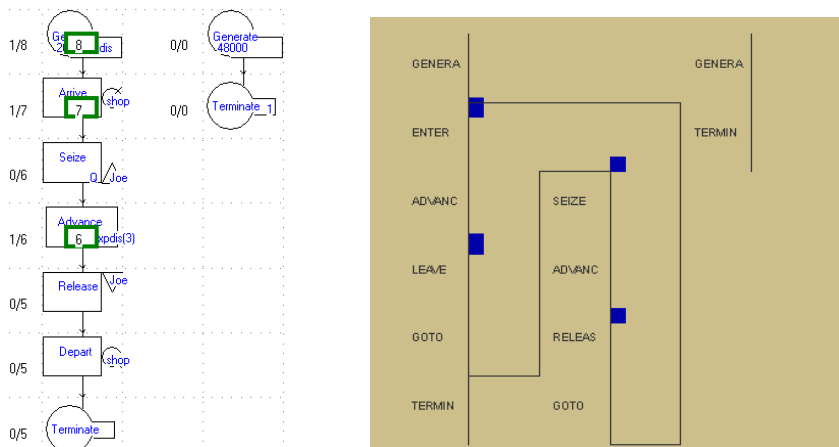

Figure 3: Block Animation in WinGPSS and aGPSS

The ideas were first discussed in Ståhl (2000), but only now, when we have the new system with free location of blocks, as presented above, is it meaningful to implement this idea. This system will automatically produce the paths of the Proof layout file to the right in Figure 3.

This animation is made from the Boris bottle shop program, discussed in the next section and illustrated there to the right in Figure 5. Clicking for Automatic animation when the block diagram of Figure 5 is loaded, both a post-run Automatic Trace File and a Layout file are automatically constructed. By loading these files into a directory with Proof Animation, the animation can be run continuously or moved forward step by step. It should be mentioned that this kind of automatic animation is made possible by the fact that the layout files of Proof are ASCII files and can hence be easily constructed by another program like aGPSS.

6   FINAL REFLECTIONS

We shall in this section briefly discuss which of the general experiences that can be drawn from our many years of teaching simulation are common to all three of us and in what regard we have different experiences. First, our goal has not been to teach a specific simulation system as such, but to give the students the main ideas of simulation, making them prepared to later move to other types of systems, both GUI and text based. In this regard, our business and high school students differ from e.g. computer science majors, who in the future might be the specialists from whom business graduates would buy more advanced simulation modeling products and/or services.

An important positive experience has been the interaction between the feed-back from the students, in form of class discussions, errors in exams and project work, on the one hand, and the continuous development of the teaching software, on the other. The system syntax has been simplified when several students have made the same mistake. Student feedback has also greatly influenced the error reporting system. This feedback has come not only from our own students in the US, Germany and Sweden, but also from students in Norway and Latvia, and lately from collaborating teachers in Japan and India. B and H have not only passed on student comments to S, but also greatly contributed with their own suggestions to make aGPSS even easier to learn and use. This development would not have been possible if we would have had a policy of constant upwards compatibility, like e.g. GPSS/H. In contrast to commercial software vendors we have not had to be concerned about a large existing customer base, since our main customers are students who are novices to simulation.

We have all three tried to ensure that students at a very early stage get to write simple programs that are not trivial. We have proceeded step-by-step, with simple examples in the beginning, so that no student is left behind at an early stage and looses the possibility to catch up. We have always avoided pre-course knowledge requirements, e.g. of computer programming. Finally, students have not had to learn a new concept every time that a new and different thing has to be done. Students have enjoyed finding that the new aspects can be handled using already known concepts, even if the programs become slightly longer.

Among other features that we have noted that students appreciate are:

- The simple and fast build up of models by using a Graphical Users Interface with the point-and-click method, allowing for a faster build-up than the drag-and-drop method.
- The automatic generation of the most interesting statistics in an understandable form, such as simple tables, histograms and graphs, under a number of different tabs.
- The automatic calculation of cumulative probabilities when defining empirical random functions. For each value, one just inputs the number of observations. aGPSS automatically transfers this into percentages and then cumulative probabilities. In contrast to other systems, the student does not have to do a lot of recalculations if one decides to add one more observation.
- The ease with which replications of the runs are done, by just one command and the automatic statistical analysis of these repeated runs, e.g. of confidence levels.

- The single easy-to-read and compact program listing, obtained automatically and allowing for short comments. This listing has been essential for making it easy for the teacher to correct and mark the student programs. This program listing, together with the block diagram, has made it easy for the students to study, discuss and document the logic of a program.
- The free, or low cost of, software. From the site www.aGPSS.com one can download, free of charge, the smallest aGPSS version, large enough for running all of the 63 programs in the text-book Born & Ståhl (2013). Many aGPSS textbooks include a software version large enough to carry out projects of a generally sufficient size. From the site http://www.wolverinesoftware.com the student version of Proof animation can be downloaded free of charge.
- The 300+ aGPSS models available, often helpful for the students when doing project work.

As teachers we have also appreciated that our GPSS classes on average have been given better ratings than other courses. The simplicity of the aGPSS syntax makes it possible for a teacher to have complete command of the whole system and to be able to answer any student questions. The clear workspace window, which is easy to project visibly to all students in a class room, facilitates teaching.

There are, however, also some differences between us as regards how we have been teaching GPSS. While B has focused on a more in depth knowledge of simulation techniques, H and S have to a larger extent been focusing on real student projects out in companies. In the classes focused on simulation, H and S have had students devote around half the time on simulation technical issues and the other half in the form of "learning by doing" in the project. In these projects, students work through the whole simulation process as regards some concrete problem with which they are familiar, like a store they worked in last summer. In this way, the students actively learn the whole process, from formulating the question to be answered, delimiting the problem system, outlining the model graphically, gathering data (e.g. on arrival and service times and on waiting lines), coding the program, verifying, validating and documenting, running the program a sufficient number of times and doing a statistical analysis for drawing significant conclusions. As regards validation, they compare data from the real system with the output data (e.g. on waiting lines) from the tentative model. If not validated, the model has to be adjusted. Finally, the students provide and test a suggestion for an improvement of the system and they present the results in a form suitable for a potential user, with a focus on implementation.

Many suggestions from these projects have been implemented. It is of interest to note that the availability of a block diagram facilitates the task of the model builder when explaining the model to the user. Successful implementation of model results requires that the user feels confident with the model, and successful models have in general been developed step by step, starting with a small model with a simple and easily understood structure. Many projects have led to small **prototype** models that have gradually been expanded in interaction with the user. Work on a simulation project in a company has in many cases landed students jobs. In many European companies, discrete event simulation is little known and many medium-sized companies have no expertise in this area. Hence, a good student can already after only twenty classroom hours do something useful in a company that no one in the company could do.

Another difference regards modeling before starting coding. H has at an early stage been focusing more on the modeling aspects. H has, when starting modeling, sometimes used video clips, e. g. of traffic intersections, and sometimes physical models. H has also used animation more than B and S.

One example of H's use of physical models, WinGPSS and Proof in the teaching of modeling in Germany can be illustrated by Figure 4 on the next page. In the class room the students study a very simple Lego railroad model, shown to the left in this figure. Their task is then to build an animation of this that is close to the Lego system. The Proof layout file might then look as in the right part of this figure. Some more details on this model are provided in Ståhl et al. (2011).
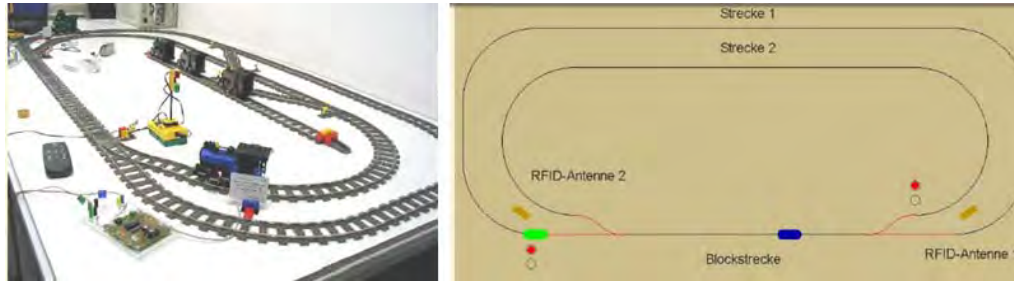
Figure 4: A Lego Train Circuit and its Proof Layout File Counter-part

It should be mentioned that B and S have now become keener on including animation, e.g. using Proof, than earlier, as witnessed by the latest development of aGPSS, discussed in the preceding section. It should, however, be stressed that this animation in aGPSS is **optional**, in contrast to many modern simulation systems like Witness, where animation is an integral part of the system, which has considerable disadvantages, as illustrated by the following experiments:

We have carried out experiments for comparing GPSS with other systems. Some have involved the following problem: "At a store, run by Boris and Naina, customers arrive at rate of 7 ± 3 minutes. A customer first goes to Boris and chooses a bottle. This takes between 3 and 7 minutes. Next he goes to Naina to pay for the bottle. This also takes 3 to 7 minutes. Finally, he returns to Boris to pick up his bottle. This takes between 1 and 3 minutes. He then leaves the store. There is one waiting line in front of Boris and one in front of Naina. A customer returning to Boris to pick up his bottle has to start at the end of this line again. The store is closed after eight hours".

One experiment has been carried out with a class of students without simulation experience. Half of them had four hours of aGPSS, the other half had four hours of a commercial system suitable for animation. At the end of each session, the students were asked to model the Boris problem in 45 minutes. While none of the students learning the animation based system could do this, all the GPSS students could do so. The reason for this difference can perhaps be explained by comparing the GPSS block diagram of this system to the left in Figure 5 to the generalized diagram to the right of how this problem can be solved in many animation based system. AD 5,2 here represents ADVANCE 5,2.
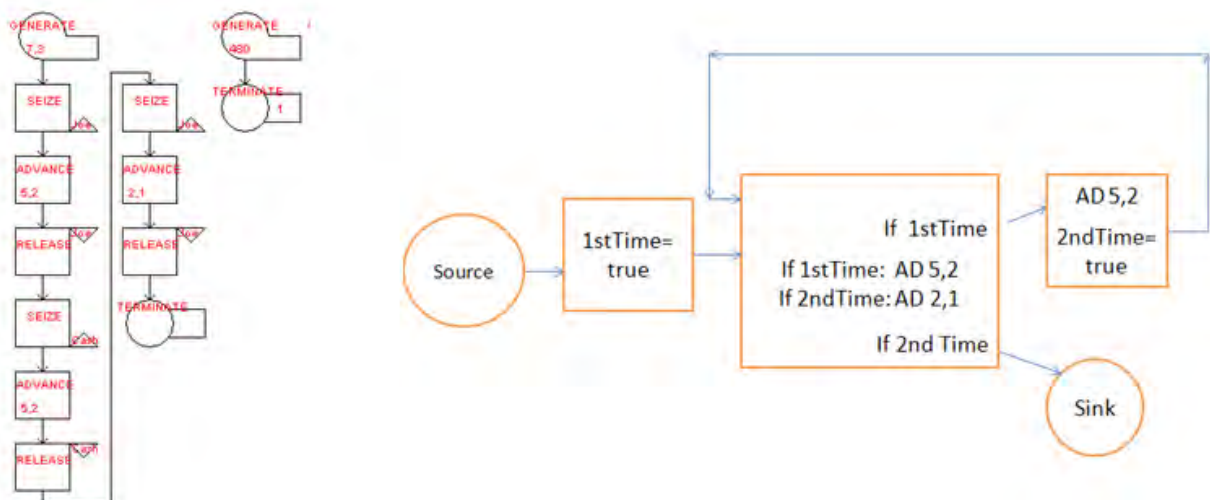

Figure 5: Diagrams of Boris Bottle Shop

In the animation based system, each permanent server is in principle only represented once, since it in the animation workspace must be in only **one** place. In the block based system, such a server can be represented in **many** different places in the model. Hence, we have SEIZE Boris in two places in the GPSS model to the left in Figure 5 and we hence do not need special logic to deal with the fact of whether a customer comes the first or the second time to Boris.

It should be stressed that in case one uses the ANIM block with Proof animation, our students will also have to deal with the more complex logic of the type shown to the right in Figure 5. The important conclusion is that we three feel that animation should be optional and only offered towards the end of the course to more advanced students.

We shall end the paper by mentioning our future plans for the aGPSS system. It should be stressed that in contrast to providers of many other systems we do **not** have the ambition to constantly enlarge, and thus complicate, our systems by introducing new features. Instead, we have during the last years deleted some features that have proved difficult to learn and have not been needed by the students, e.g. in their work out in companies. Our most urgent priority is rather to focus on providing many more examples of applications, showing that one with our simple aGPSS system can cover a great amount of application areas. We shall especially focus on applications in business, since we believe that simulation should play an important part of a business school curriculum.

Among the issues that we are working on, one can mention the establishment of optimal equipment life, overbooking and price differentiation in the airline industry, bidding on stocks of an oil company with uncertainty about the success of an oil exploration process and a model of stocking of perishable goods in a supermarket. We also want to illustrate some basic ideas of continuous simulation as applied to business problems with the aim of doing market forecasts and evaluations of high tech stocks. We also want to deal with the evaluation of European stock options for the case of volatility varying over time, simulation based costing and the use of laptop based simulation models for sales support.

REFERENCES

aGPSS Demo version. 2013. Last modified May 5 2013. http://www.aGPSS.com.

ASIM. 1993 *GPSS-User's Group Europe. Gründungsveranstaltung*. Heft NB 36.

Born, R. 1988. *Processor potential utilization in very-large-scale, regular, static interconnection networks*. Ph.D. dissertation. Illinois Institute of Technology, Chicago.

Born, R. 1991. "Simulation as a Tool for Telecommunication Network Modeling in an MIS Data Communications Course." *The Journal of Computer Information Systems*, Spring 1991.

Born, R. 1997. *The Practical Power of Spreadsheet Simulation Modeling*. Working Paper. Northern Illinois University.

Born, R. 2003. "Teaching Discrete Event Simulation to Business Students: The Alpha and Omega." In *Proceedings of the 2003 Winter Simulation Conference,* Edited by S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 1964-1972. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Born, R. and J. Kenevan, 1988. "Closed Form Derivation of Processor Potential Utilization in Straight Line, Ring, and Hypercube Networks." *CMG Transactions,* Winter 1988.

Born, R. and I. Ståhl. 2004. *The First Hours of Simulation Education*. Stockholm: SSE.

Born, R. and I. Ståhl. 2004b. "The First Hours of Simulation Education". In *Proceedings of the 2003 Winter Simulation Conference,* Edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 2066-2074. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Born, R., I. Ståhl and H. Herper. 2005. "Die ersten Stunden in der Simulationsausbildung." In *Modellbildung, Simulation und Animation in der schulischen Ausbildung. 1. Magdeburger Lehrertag Informatik.* Impara, Magdeburg.

Born, R. and I. Ståhl. 2007. *Simulation Made Simple with WebGPSS*. Gothenburg: Beliber. 2007.

Born, R. and I. Ståhl. 2013. *Modeling Business Processes with a General Purpose Simulation System – 1. Introduction.* Stockholm: SSE.

Frank, M. and P. Lorenz. 1979. Simulation diskreter Processe. Leipzig: VEB Fachbuchverlag.

Herper, H. and I. Ståhl. 2003. "Modeling and Simulation in High School Education - Two European Examples." In *Proceedings of the 2003 Winter Simulation Conference,* Edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 1973-1981. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

MacMillan, C. and R. Gonzales. 1973. *Systems Analysis - A Computer Approach to Decision Models.* 3rd ed. Homewood, IL: Irwin.

Proof Animation Student version. 2013. Accessed August 6, 2013. http://www.wolverinesoftware.com.

Schriber, T. 1974. *Simulation Using GPSS*. NY: Wiley.

Ståhl, I. 1990. *Introduction to Simulation with GPSS - on the PC, Macintosh and VAX.* Hemel Hempstead, UK: Prentice Hall International.

Ståhl, I. 1991. *Simulation in One Day with micro-GPSS.* A Short Tutorial. Stockholm: SSE.

Ståhl, I. 1992. "Animation with micro-GPSS and Proof." In *Visualisierung und Präsentation von Modellen und Resultaten der Simulation,* ASIM, Heft. Nr. 31.

Ståhl, I. 1996. *Simulation Made Simple with micro-GPSS. A Short Tutorial with Eight Lessons*. Stockholm: SSE.

Ståhl, I. 2000. "Automatic Animation with a Block Based Simulation Language." In T. Schulze, P. Lorenz and V. Hinz (eds.) *Simulation und Visualisierung 2000*. Erlangen: SCS & ASIM.

Ståhl, I. 2003. *Simulation Made Simple with WebGPSS – A Tutorial*. Stockholm: SSE.

Ståhl, I. and H. Herper. 1998. *Einführung in die Simulation mit Micro-GPSS*. SSE and Universität Magdeburg.

Ståhl, I. and H. Herper. 1999. "Micro-GPSS on the Web and for Windows: A Tool for Introduction to Simulation in High Schools." In *Proceedings of the 1999 Winter Simulation Conference*. Edited by P. Farrington, H. Nembhard, D. Sturrock and G. Evans, 298-306. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ståhl, I,. J. Henriksen, R.Born and H. Herper. 2011. "GPSS 50 Years Old, but Still Young." In *Proceedings of the 2011 Winter Simulation Conference.* Edited by S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu. 3952-3962. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Watson, H. and J. Blackstone Jr. 1989. *Computer Simulation.* 2nd ed. NY: Wiley.

AUTHOR BIOGRAPHIES

INGOLF STÅHL is professor emeritus of the Stockholm School of Economics, but continues his work on aGPSS and its application to business problems. His email address is ingolf.stahl@hhs.se and his aGPSS system is at http://www.aGPSS.com.

RICHARD G. BORN is an associate professor emeritus of Northern Illinois University, but continues teaching simulation modeling of business processes and his work with Ståhl on aGPSS. His email address is rborn@niu.edu.

HENRY HERPER is at the Institute for Simulation and Graphics at the Otto-von-Guericke University, Magdeburg. His research interests include the development and evaluation of simulation tools and methodology for simulation education. His email address is henry.herper@ovgu.de.