Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

GPSS - 40 YEARS OF DEVELOPMENT

Ingolf Ståhl

Department of Managerial Economics
Stockholm School of Economics
Box 6501
SE-11383 Stockholm, SWEDEN

ABSTRACT

This year GPSS celebrates its 40th birthday. This paper reports on the development during these 40 years, starting with the first version developed by Gordon at IBM in 1961, and the following development of GPSS II, GPSS III, GPSS/360 and GPSS V, all IBM products. A major section is devoted to GPSS/H, which has dominated the GPSS scene during the last years. There is one section on the GPSSR family of GPSS versions and one on GPSS/PC and GPSS World. There are also many GPSS systems, projects and ideas of a mainly academic nature. A great number of GPSS textbooks are noted. The concluding section discusses the reasons for the popularity of GPSS.

1 INTRODUCTION

This year GPSS will celebrate its 40th birthday. In 1961, IBM released the first version of GPSS, developed by G. Gordon. It is remarkable that 40 years later intensive development on GPSS still continues and that new versions of GPSS have recently been released, like GPSS World for Windows and WebGPSS for the Web. I shall try to bring out some of the most prominent features of the different systems developed during these 40 years. Since there is an great amount of differences, I shall as a representative measure of complexity of the different systems use the block types, in particular the total number of such types.

2 THE FIRST GPSS VERSIONS

The early history of GPSS has been told by Gordon himself in a well-known paper in ACM SIGPLAN Notices (Gordon 1978). This is the only source on the development of GPSS up to 1961. From this source we can learn that the earliest ideas that eventually resulted in GPSS came to Gordon in the late fifties when he was working at Bell Telephone Laboratories. Already in his earliest attempts of doing simple simulation of message switching systems he used single-server and multi-server entities as basic elements, the precursors of facilities and storages. In 1959 he worked on a simulator based on sequence diagrams for studying switching systems. In a paper describing this simulator (Dietmeyer *et al.* 1960) the system was described in more general terms, including people in a supermarket.

In June 1960 Gordon joined IBM. He here became involved in the study of problems like response time in terminal systems. The potential value of simulation was recognized in the department dealing with this study and Gordon started to construct a block-based simulator, partly based on his experience with the mentioned sequence diagram simulator. He was initially the only programmer. A break-through came when he in a very short time could make a simulation of a random access disk system. In October 1960, an IBM internal manual for this simulator was produced. The system had not yet got a name, but was within IBM called the Gordon Simulator. Certain anomalies in the way the simulator handled simultaneous events were discovered and Gordon redesigned the simulation algorithm. For the next events, taking place 40 years ago, I shall quote Gordon (1978), p 186-187, directly:

"At the beginning of 1961, a complete rewrite of the program was begun in order to implement the new algorithm that was developed. … I was joined by two programmers, Mssrs. R. Barbieri and R. Efron, and the program continued through the summer of 1961. … The confidential classification of the program was removed early in 1961 so that copies of the then existing program could be given to people outside IBM, on the understanding that is was not supported by IBM. (It was an IBM III program, for which there is no commitment to correct errors.)

It took a few more months to stabilize the program and produce proper documentation. When this was done, a press release was made on Sep. 27th 1961, announcing the program as an IBM type II program (that is, with a commitment to correct errors.) A user's manual, dated October 6th 1961 was issued, and the program was made available for use on the IBM 704, 709, and 7090 systems. (Gordon 1961) ... in the haste of preparing for its release it was just called General Purpose Simulator, or GPS. I shortly realized that this acronym was used by Professors Newell and Simon for a program called General Problem Solver. … I quickly changed the name to a 'General Purpose Systems Simulator', or GPSS."

A paper describing the program was presented at the Eastern Joint Computer Conference (Gordon 1961b). An article by Gordon of roughly the same size, but more readily available, was published in IBM Systems Journal in 1962 (Gordon 1962). From the three above-mentioned sources, we can learn the following details about the first version of GPSS. There were altogether 25 block types: ORIGINATE, GENERATE, CALL, ADVANCE, BRANCH, SEIZE, RELEASE, HOLD, INTERRUPT, PREEMPT, RETURN, ENTER, LEAVE, STORE, QUEUE, SPLIT, MATCH, ASSIGN, MARK, TABULATE, TAG, TRANSFER, GATE, WRITE, TERMINATE (see also the Appendix). There were several blocks that are quite different from the blocks in later GPSS versions. For the sake of the later discussion, we shall look somewhat closer at some of these differences.

There were two blocks for creating transactions: ORIGINATE and GENERATE. The creation of transactions from ORIGINATE would continue even if the transactions were unable to enter the next block, while it with the GENERATE block would continue only so long as the transactions can move to the next block. It is here of interest to note that later IBM versions of GPSS as well as GPSS/H had the GENERATE block operate like this original GENERATE block, while other GPSS systems, like GPSS/PC and micro-GPSS, had their GENERATE block behave like ORIGINATE of this original GPSS.

Already in this first version of GPSS there were two types of service stations, "facilities" giving service to one transaction at a time and "stores" (corresponding to storages in later versions) that are able to give service to several transactions at a time. The facilities could be seized and freed by SEIZE and RELEASE and a unit of a store by ENTER and LEAVE. The determination of the service times for these servers were, however, different in this first version compared to present versions. There were three different blocks for delays. The ADVANCE block was used only to represent time that did not involve the use of a service station, while HOLD was used for times that the transaction used a facility and STORE for times that the transaction used (part of) a store.

The following feature involves probably the biggest difference compared to all present versions of GPSS: All block types other that those representing terminal points could have two exits, exits 1 and 2. They were defined by giving the number of the block to which the exit would lead. If there was a unique exit, only one number was given, as exit 1. If there were two exits, both exits had to be defined as well as a selection factor to determine the choice between the exits, like the probability of going to exit 1. The coding of the operands was based on a fixed format where for example exit 1 started in one particular column and the selection factor in another specific column.

In general, there was a great set of limitations in this very first version of GPSS, like only one random number stream, only one parameter and only numeric representation of the servers. There were no Standard Numeric Attributes and the GATE block could in the very first version of GPSS only refer to server numbers. It can finally be noted that the capacity of a store was defined using a statement called CAPACITY, just like in one of the most recent versions of GPSS, WebGPSS.

The next version of GPSS, **GPSS II**, did not provide a radical change from the original GPSS; (IBM 1963) and (Efron and Gordon 1964). The total number of block types increased, however, from 25 to 33. 12 new blocks were added (ASSEMBLE, BUFFER, COMPARE, HELP, INDEX, LOGIC, LOOP, PRINT, PRIORITY, SAVEX, TRACE and UNTRACE), while 4 old ones were deleted (BRANCH, CALL, TAG and TRANSFER). A new VARIABLE statement allowed algebraic computations upon the system variables. Also other features increased the power of GPSS: The number of parameters increased to 8. The parameters could be used also for indirect addressing. The user could call on user-supplied subroutines written in assembly language. Also a simple form of macros was introduced. Output statistics were expanded.

A much more radical change came with the release of **GPSS III**. Although the number of block types increased only by three to 36, there is a bigger closeness between e.g. GPSS World of 2000 and GPSS III than between GPSS II and GPSS III (Herscovitch and Schneider 1965). The main changes involved the following: The ORIGINATE and GENERATE blocks were consolidated into one single GENERATE block, mainly working like the original ORIGINATE block (see above). The HOLD and STORE blocks were eliminated by allowing ADVANCE to be much more general. The two-exit system with its selection factor was eliminated. The selection of the next block was instead determined by the TRANSFER block, which reappeared. The fixed format with each block operand having to start in a specific column was replaced by a format where all the operands came in one uninterrupted string.

Furthermore, a DEPART block was added to work with the QUEUE block. The two earlier versions of GPSS had only had a QUEUE block, measuring the queue when placed in front of e.g. a SEIZE or an ENTER block. This single QUEUE block was, however, unsuitable when measuring some other queues like in front of a pair of GATE blocks. With the new QUEUE/DEPART block pair, such a queue could be measured. LINK and UNLINK blocks were added to allow for user chains, supporting in certain cases more efficient execution. SAVEVALUE replaced SAVEX. TEST replaced COMPARE. Other improvements were a larger number of parameters, a more general PRINT block allowing for the print-out of any SNA, not just savevalues (then called savexes) as in the two earlier versions. The SPLIT block now allowed for more than one copy.

In connection with the GPSS III release, the GPSS acronym was changed to mean General Purpose Simulation System (rather than Systems Simulator). With the advent of GPSS III, GPSS became more clearly a simulation language than a simulator.

GPSS III was succeeded in 1967 by **GPSS/360** (Gould 1969), which just implied new extensions, e.g., in the form of 8 new block types, bringing the total to 44. Of the 8 new block types 5 dealt with a new kind of entity, the group: ALTER, EXAMINE, JOIN, REMOVE and SCAN. The other three new block types were COUNT, MSAVEVALUE and SELECT. Other extensions introduced into GPSS/360 were introduction of parameters and savevalues of different lengths as regards number of bits used (16 and 32). Earlier there had just been one type of parameters (16) and one type of savevalues (savexes: 32). QUEUE/DEPART pairs were now also allowed to include time spent in ADVANCE blocks, implying that one with this kind of pair could measure the time spent by a transaction between any two blocks, implying that the QUEUE block should now rather be called ARRIVE. Now 8 different random number generators were introduced. The VARIABLE concept was extended to include parentheses and Boolean variables. An important change was that now any entity, not only blocks, could have symbolic names, in addition to their numeric ones. PREEMPT was extended, e.g. to allow for regard to priority. The macro capability was significantly extended. Finally, an output editor allowed the user to specify the format of the output.

GPSS/360 was, in turn in 1971, succeeded by **GPSS V**. Four new block types were introduced regarding the availability of facilities and storages: FAVAIL, FUNAVAIL, SAVAIL and SUNAVAIL. Thus, the total number of block types reached 48. A capability to place part of the model on external storage was introduced. Parameters and savevalues could now take four different forms. In contrast to earlier editions, a floating-point format was allowed.

At this time, several other GPSS versions had already been introduced by manufacturers other than IBM: CDC, Honeywell, Norden, RCA, UNIVAC and Xerox (Gordon 1978). The Norden-360 version, developed by a group led by J. Reitman, was innovative in terms of memory usage, interactive input and use of graphic displays on terminals of simulation output (Nance 1996). In Germany, Siemens and Telefunken had their GPSS versions.

3 GPSS/H

The GPSS versions mentioned above, were all run on mainframes and the GPSS versions were all, with the exception for the Norden version, developed by computer manufacturers. In the late 70's and early 80's there was, due to the policy of unbundling and the emergence of the minicomputers, a movement towards software being developed by independent software houses. This was also the case of GPSS. Most noteworthy of the GPSS versions developed by independent software firms is GPSS/H. It has no doubt during at least the last decade been the GPSS software with the largest use. Against this background, it appears reasonable to give a more detailed review of GPSS/H than of other GPSS versions.

The history of GPSS/H started with J. Henriksen taking T. Schriber's course in GPSS at the University of Michigan in 1968. At this time, a major critique of the GPSS versions then used (mainly GPSS/360) was the slow execution. Working with GPSS/360 at the Computing Center of the University of Michigan, Henriksen got ideas about how to make a much faster GPSS. The main key to increasing the speed of execution was to make GPSS fully compiled, instead of mainly interpreted, which had been the case with all the IBM versions. GPSS/H was completely upwards compatible with GPSS V, but as mentioned much faster. Henriksen started Wolverine Software in 1976, roughly the same time as IBM stopped supporting GPSS V.

Several blocks were added to those of GPSS V. When the first GPSS/H was released in 1977 it had a total of 58 block types. This has over the years increased (62 in the 2nd version of 1988 and 68 in the 3rd version of 1995). The present version of GPSS/H has a total of over 70 block types. Another important extension has concerned control statements, allowing among other things for looping and hence simple run control. A noteworthy difference to GPSS V is the handling of several repeated runs, in GPSS/H requiring far fewer statements than in GPSS V. In contrast to GPSS V, which allowed for names of at most 5 characters, GPSS/H allows names of up to 8 characters, provided they do not conflict with reserved words. The only major difference as regards the result when running old GPSS V programs is that GPSS/H has floating-point clock values, while GPSS V (as well as all earlier IBM versions) had integer values. This will lead to fewer events in GPSS/H scheduled simultaneously and fewer cases of unexpected results due to truncation of clock values.

GPSS/H has in a number of comparative test runs proved to run a lot faster than GPSS V, by at least a factor of 5; (Henriksen 1983) and has also for similar programs in other simulation languages proved to execute several times faster (Abed *et al.* 1985). Another significant improvement compared to GPSS V was a new interactive debugger, allowing for "just in time" debugging. GPSS/H has been gradually improved in many directions, also in the 90's. In 1993, a Runtime version of GPSS/H was released, allowing the running of precompiled version of GPSS/H program, and in the third version of GPSS/H from 1995, 23 new statistical distributions were introduced.

The compiler of the first version of GPSS/H was written in assembly language for the IBM main frames. In the 80's with the release of GPSS/H versions for several kinds of minicomputers like the VAX, the compiler was rewritten in C. In 1988 a version for the PC (MS/DOS) was released. It should be mentioned that GPSS/H is very similar across all platforms. There is no Windows version nor any GUI version. The development efforts of Wolverine have instead been focused on SLX and Proof Animation. GPSS/H is described in manuals, and in some textbooks, by Banks and others (1989) and by Schriber (1991).

Because of GPSS/H's dominating position in the GPSS area and due to its open structure, with all program input and output being in the form of ASCII text, a great amount of software has been developed that connects to GPSS/H, both at the front-end, producing (parts of) programs, and at the back-end, producing better, more readable output. I shall here mention a few of these programs. In the 1980s two software packages both with animation capabilities were presented to be used with GPSS/H: TESS (The Extended Simulation System from Pritsker & Associates) and AutoGram of AutoSimulations. These animation packages became uninteresting for GPSS/H users with the advent of Wolverine's own Proof Animation at the start of the 90's.

AutoMod, also from AutoSimulations, was initially a pre-processor that automatically generated GPSS/H program code. In the 80's researchers also sought to use RESQ as a front-end for GPSS/H, (Mathewson 1989). In 1993, R. Elnicki presented a program RUN as a fast front–end or "shell" for running GPSS/H on PCs (Wolverine 1993). Also in the early 90's, Mogul of High Performance Software was used for generating GPSS/H code for communication systems, (Rodrigues 1993). In the early 90's, a German firm, GfL in Aachen, released GPSS/H EDITOR, a front-end for inputting GPSS/H programs mainly by clicking on buttons with text for blocks, but without a true Graphical Users Interface, i.e. without a block symbol menu or block diagram (Knepper and Krönchen 1993).

Since 1994 a copy of UniFit II is bundled with every commercial GPSS/H Professional version, allowing the user to determine the best probability distribution for the data. The mid 90's also saw the release of SIMSTAT (from MC^2 Analysis Systems), which reads and analyzes output generated by GPSS/H (Crain 1996).

4 THE GPSSR FAMILY

What I shall here, for the sake of brevity, call the GPSSR family of GPSS dialects was developed at the University of Western Ontario, mainly under the leadership of D. Martin, and then made available through the firm Simulation Software. The following information is based on (Richards 1987 and 2001).

The first product released was GPSS10, developed for DEC PDP-10 and written in assembly language and BLISS10. It was mainly based on GPSS V. One advantage over GPSS V was that it had an interactive debugger. This was in 1980 followed by GPSS/VX for DEC VAX/11, written in BLISS32. It had compiled expressions and an enhanced interactive debugger. 1981 saw the release of GPSSR for DEC PDP-11, written in C. Among the new features one can mention that SNAs could be used in any field, including FUNCTION definitions. A restriction was that the simulator and the report generator were separate executables due to limited memory. In 1982, one released a system called GPSS/C, which was developed to be easily ported to any architecture, since the GPSS code was converted into P-Code and evaluated with an interpreter. The GPSS/C system was during the next decade enhanced and ported to over a dozen different machines.

In 1983, the first GPSS on the PC, GPSSR/PC, was released. It was developed by B. Richards. Compared to GPSSR, it had added functionality, e.g. in the form of an interactive report generator. It had a total of 42 block types. (7 of these are not in any system in the Appendix.) Later some animation capabilities were added.

5 GPSS/PC AND GPSS WORLD

In 1984, one year after the release of GPSSR/PC, Minuteman Software, under the leadership of S. Cox, released a PC version of GPSS. This was software that was directly developed for the PC, without any predecessors for larger computers. The syntax was mainly built on that of GPSS V, but it was to some extent a subset, with e.g. the block types CHANGE, HELP, PRINT and WRITE left out, implying a total of 44 block types. It had a built-in editor to some extent resembling that of the popular BASIC editors at that date, with a line number in front of each line to facilitate insertion. The editor had a built-in syntax check. A new feature was built-in graphic displays of key statistics updated dynamically as well as a very rudimentary form of animation in the form of a display of the transactions moving through the block diagram. Names of up to 20 characters were allowed, provided they were not reserved.

Like GPSS V, but unlike GPSS/H, it had an integer clock, but now with almost infinite size. Also the other SNAs were of integer type, but with almost unlimited internal precision. Unlike both GPSS V and GPSS/H, in GPSS/PC arrivals of new transactions in a GENERATE block were not stopped by a blocking condition in the block following the GENERATE block, but transactions can wait in the GENERATE block. Due to the memory restrictions of DOS, there was a division between the animator and the report generator. GPSS/PC was not compiled but mainly interpreted. For similar programs run on a PC, execution was slower for GPSS/PC than for GPSS/H.

In 1988 GPSS/PC Animator, a DOS-based post processing system that supported the generation of 3D animations on the basis of AutoCAD drawings was released. 1988 also saw an improved EMS version of GPSS/PC allowing increased memory usage. Around 1990 several textbooks based on GPSS/PC and including its student version on a diskette appeared, (Karian and Dudewicz 1991), (Chisman 1992) and (Thesen and Travis 1992). A more fundamental improvement was announced in 1994 in the form of GPSS World. This was first developed for OS/2. This OS/2 version of GPSS World was announced to be accompanied by an animation program, Simulation Studio, allowing for 2½ D vector based animation. Mainly due to the poorer acceptance of OS/2 than of Windows, the development efforts of Minuteman turned to Windows and in 2000 a GPSS World for Windows was released.

The new GPSS World is upwards compatible with GPSS/PC except for animation. There are many new features in GPSS World. There are 9 new block types, bringing the total to 53. The new block INTEGRATION is meant to facilitate continuous and hybrid simulation. Other new blocks allow for flexible file handling. GPSS World has a completely different visual appearance than GPSS/PC. It has a full screen editor of traditional Windows type. For the display of results it uses a total 20 different windows. The presentation of the transaction movements through the block diagram is, however, less graphical than in GPSS/PC, but is more suitable for debugging. It also has a new much faster translator. The SNAs take floating-point values instead of only integer values as in GPSS/PC. GPSS World includes PLUS, the Programming Language Under Simulation. PLUS expressions can be included almost anywhere in the GPSS program, either in blocks or in procedure calls, thereby increasing the power of the programs that can be written. PLUS allows program control based on simulation results. GPSS World allows multitasking, allowing several simulations to be run simultaneously.

It should be mentioned that student versions of both GPSS World and GPSS/PC can be downloaded from www.minutemansoftware.com.

6 OTHER GPSS LANGUAGE VERSIONS

In the late 70's, there were also other independent developers of GPSS software or ideas for GPSS, not only in North America, but also Europe. These GPSS versions or projects most often originated at universities. Much of what I report on in this section concerns projects and is based on articles in journals and proceedings, with at least a partly academic focus. It is in general impossible to know how successful these projects have been. My main purpose in reporting on them is to show the general ideas, as well as to give a picture of the great amount of work done on GPSS.

The earliest work to report on is from the mid-80's, namely the development in Magdeburg, Germany, on SIMDIS, a GPSS V similar version for IBM similar mainframe computers. It would eventually include extensions to GPSS V, e.g. data base capabilities, (Preuss 1987). On the basis of SIMDIS, a PC-version of GPSS, called SIMPC, was developed in Turbo-Pascal around 1990. This version had several extensions compared to SIMDIS, e.g. a built-in editor and extended reporting concurrent with execution (Schulze 1991). The 1990s saw also another German PC-based GPSS version, developed in Turbo-Pascal, called GRAMOS-GPSS (Diedenhofen 1993). Its main feature was a simple GUI, where the user could build the programs by selecting blocks from a menu of 40 GPSS block symbols.

In the late 80's, a PC-based GPSS, called micro-GPSS, was also developed by I. Ståhl in Sweden (Ståhl 1990 and 1996). It was intended to be used in education. The development was gradual, based on feedback from over 5000 students in Sweden and the US. It started out as a pure subset of GPSS V, but under student influence, the system was greatly simplified, so that teaching material that with GPSS V required 22 hours with later micro-GPSS versions only needed 10 hours. Some of the many changes from GPSS V are: Only floating-point values, transactions can wait in GENERATE (e.g. prior to a SEIZE block), a straight-logic IF replaced TEST, WAITIF replaced GATE, ARRIVE is used instead of QUEUE, a simpler GOTO replaced TRANSFER and LET replaced ASSIGN and SAVEVALUE. In total there are only 22 block types, i.e. fewer than in the original GPSS of 1961. Yet, 99 percent of all programs in the GPSS textbooks below have been rewritten in micro-GPSS with at most 20 percent longer code. It should be mentioned that the PRINT and HELP blocks are more powerful than in GPSS V. The HELP block is e.g. used for making graphs and for a simple interface with Proof Animation. Micro-GPSS is written in very portable FORTRAN and has been ported to Macintosh, VAX, SUN and LINUX systems. Although micro-GPSS is not compatible with GPSS/H, there is, as a program GPHM by which micro-GPSS programs can be translated into GPSS/H code.

In the 90's, there have been several efforts regarding the development of a GUI for making the build-up of a GPSS program very simple by clicking on symbols in a symbol menu. The first effort towards such a GUI for GPSS was made by Ball (1992) for a new GPSS dialect GPSS/VI, with VI for visual. This was originally for the Macintosh. This has been followed by two projects of a GUI for micro-GPSS. A Windows based version, WinGPSS, with extensive HELP pages, is now being developed in Magdeburg by H. Herper, A. Krüger and H. Schliefke (Herper and Ståhl 1999). A GUI version for the Web, WebGPSS, was released in 1999. Financed by the Swedish KK-foundation, it was meant to be used also in high schools. The system is restricted to the 16 most fundamental block types (Ståhl and Hall 1999).

7 SUGGESTIONS FOR A FUTURE GPSS

Criticism of certain aspects of GPSS syntax has come not only from adherents of competing systems, but also from within the GPSS community. Thus, for example, both Gordon (1979) and Henriksen (1983) raised the objection that, due to the gradual expansion of GPSS and the adherence to upwards compatibility with earlier versions, GPSS had got more and more statements doing roughly the same thing. Some kind of consolidation was necessary. Among other critical aspects raised was the lack of a possibility of making truly modular programs, since GPSS could not give the programs a hierarchical structure.

Against this background, Henriksen (1985) outlined ideas of a new GPSS in a paper on a "GPSS/85", where the concept of extensibility is brought into focus. It appears that in the further development of such a new GPSS, Henriksen gradually found it more and more important to start from scratch, with a new language based not only on GPSS, but also on C. This new language, called SLX, was first made public in 1993 (Henriksen 1993). It can be regarded to consist of several layers. The bottom layer, involves very few concepts, but has a very powerful and efficiently implemented **wait until** concept. On a layer above is a module of statistical procedures defined in this base language and on a layer even higher a module for SLX definitions for GPSS/H-like entities, defined by use of the base language and the statistics module. While the base language contains the GPSS words **advance** and **terminate**, the GPSS/H like module contains GPSS words like SEIZE, RELEASE, ENTER, LEAVE, QUEUE and DEPART.

However, SLX has, even on this GPSS-like level, some clear differences compared to GPSS. There is no GENERATE concept to be placed at the top of a segment, e.g. prior to QUEUE. Furthermore, it would be very cumbersome to produce the block statistics of GPSS in SLX. In fact, it appears to be a safe prediction that there will never be a translator that will take any GPSS/H program and translate it automatically into SLX code. Hence, SLX cannot really be regarded as just an advanced version of GPSS.

Other GPSS-developers have wanted to keep the upward compatibility with earlier versions. Behlau, Hinz and Lorenz have in two papers (1993) discussed suitable extensions of GPSS/H, involving sub-models to allow a hierarchical structure in GPSS, but also a preamble part in which e.g. transactions, stations and queues are defined, possibly in vectors. Similar ideas about a hierarchical GPSS formed, at roughly the same time, the basis for the development of HGPSS in Belgium. As a first step towards modularity, the program is, just like for SIMAN, divided into a model part and an experimental part. The model part can in turn be divided into sub-models, possibly parameterized, which in turn contain other sub-models. The models are translated into C++ code, which is then compiled and executed. C++ code can also be directly inserted into the code and new blocks can be defined in C++ (Clayes *et al*. 1995).

8 GPSS SIMILAR SYSTEMS

The general structure, as well as some keywords, of GPSS has also inspired more indirect software development work. Already in the 1970s, there was an emergence of GPSS similar packages written in general programming languages: GPSSS, based on Simula, by Vaucher (Vaucher 1975), APL-GPSS (IBM 1976), GPSS-FORTRAN (Schmidt 1979), Pascal-based PASSIM, (Uyeno and Vaessen 1980], PL/I GPSS (IBM 1981) and PROLOG-based PROSS (O'Keefe 1989). This type of development has continued up to our days, e.g. with projects on a Java based GPSS (Beikirch 1997). There have also been projects involving the incorporation of GPSS-ideas into an application package, like MATLAB-GPSS (Pawletta *et al.* 1998). SLX could also have been included here.

9 GPSS TEXT BOOKS AND CONFERENCES

One must also point at the great number of GPSS textbooks. The most famous one is Schriber's classic Red book of 1974, but books by Gordon (1969) and (1975), Reitman (1971), Greenberg (1972), MacMillan and Gonzales (1973), Bobillier *et. al*. (1976), O'Donovan (1979), Solomon (1983), Banks *et. al*. (1989), Watson and Blackstone (1989), Ståhl (1990 and 1996), Schriber (1991), Karian and Dudewicz (1991), Chisman (1992), Thesen and Travis (1992), Ståhl (1997) and Silverman (1997) are examples of other simulation books in English that are mainly based on GPSS, or at least have a substantial presentation of GPSS. There have also been several text book on GPSS in German, e.g. those by Frank and Lorenz (1979), Roesmann (1979), Weber *et al* (1983), Steinhausen (1994) and Ståhl and Herper (1998). Examples of the many GPSS books in other languages are Stefanov (1975), Seprödi (1980), Avens-Aventiņš *et al*. (1987) and Yackiv *et al*. (1994).

Furthermore, reference manuals on GPSS/H, GPSS/PC, GPSS World and WebGPSS have continuously been updated. There are also several courses based on GPSS available on the Web; see e.g. Lorenz and Schriber (1996) and webgpss.hk-r.se.

During these 40 years, there have been a great many applications using GPSS. Many of them have been reported at the annual conferences now known as the Winter Simulation Conferences. It is here important to note that the very first one in this series of conferences was named the Conference on the Applications of Simulation Using GPSS, held 13 –14 November 1967.

10 CONCLUDING REMARKS

I have above shown that there has been a lot of development activities as regards GPSS over these 40 years. It should also be mentioned that GPSS has been very widely used in practice. For many years it has indeed been the most widely used one of all systems for discrete-event simulation. This was clearly the case in the beginning of the 80's (Christy and Watson 1983), but still in the mid-90's GPSS was, at least according to one investigation (McHaney 1996), the most widely used system.

Against this background, the question arises, what the reasons are that GPSS has been able to prevail for these 40 years and inspired all these efforts and this extensive usage. It is most probably not sufficient to say that GPSS is one of the oldest simulation languages. The oldest one, GSP, the General Simulation Program, developed by Tocher around 1958 (Tocher and Owen 1960), has not to any extent the same "track-record" as GPSS. Instead, the continued popularity of GPSS has to do with the most basic characteristics of GPSS, which have remained over the years. I have discussed these characteristics in detail elsewhere, (Ståhl 1993), so I shall here only try to summarize.

GPSS is in contrast to more programming language oriented simulation languages like Simula, more problem oriented. The transaction flow model with temporary transactions moving through the system to be served by permanent servers is natural for the main part of simulation tasks. GPSS, in particular in its most recent forms, is very easy, and fun, to learn. Students can after a short time construct model of some complexity. The automatic provision of statistics is helpful for the novice. For many real systems, modeling in GPSS is also much easier than in an Animation Oriented System (AOS), like e.g. Witness. In an AOS, each permanent server, like a machine, is represented only once, since it in the animation workspace, representing e.g. the factory floor, must be in only one place. In GPSS a certain machine can be represented in many different places in the program. Only a system where each product has its own machines, each visited only once, is as easy to model in an AOS as in GPSS. If each machine, however, processes more than one product, modeling is simpler in GPSS than in an AOS.

The GPSS block diagram with very readable block symbols, many of which come in clearly distinguishable pairs, makes it easier to discuss the model logic with the users. The compact programs and the possibility to build up the model using a GUI makes it easy to rapidly produce a prototype model on which one can get feed-back from the user for improvements. These last factors make it easier to implement the simulation model in practice. The compact GPSS programs, together with the block diagrams, provide in themselves better documentation than is possible with most other simulation systems, in particular AOS.

Execution efficiency in GPSS can generally be quite high, due to the efficiency of GPSS/H and the fact that programs in some other GPSS versions can be automatically translated into GPSS/H code. There is also considerable discussions in the literature on suitable ways of writing GPSS code to increase execution efficiency; (Born 1998), (Bobillier *et al.* 1976) and (Ståhl 1998).

New GPSS versions are generally better debugged than other kinds of new simulation systems, since runs in the new GPSS systems can be compared with runs in old GPSS systems. GPSS versions have also been subject to more scientific scrutiny than other simulation systems (Schriber and Brunner 1998). GPSS finally has the advantage that there are probably over a thousand different models in GPSS from a very broad area documented in the literature. A GPSS user beginning to model a problem will hence often have a good starting point in an old program.

Because of these and other advantages of GPSS, I am certain the GPSS will prevail also in the future and that GPSS will be a major player in the simulation area also when it reaches its golden anniversary in 2011.

ACKNOWLEDGEMENTS

This paper has benefited greatly from comments and/or documents from B. Richards, S. Cox, J. Henriksen, H. Herper, P. Lorenz, R. Sargent, T. Schriber, T. Schulze, D. Stanford and J. Tolujevs.

APPENDIX

Block types of various GPSS versions

	I	II	III	6	V	P	μ		I	II	III	6	V	P	μ
ADVANCE	x	x	x	x	x	x	x	LOOP		x	x	x	x	x	
ALTER				x	x	x		MARK	x	x	x	x	x	x	
ARRIVE							x	MATCH	x	x	x	x	x	x	
ASSEMBLE		x	x	x	x	x	x	MSAVEVALUE				x	x	x	
ASSIGN	x	x	x	x	x	x		ORIGINATE	x	x					
BRANCH	x							PREEMPT	x	x	x	x	x	x	x
BUFFER		x	x	x	x	x		PRINT		x	x	x	x		x
CALL	x							PRIORITY		x	x	x	x	x	x
CHANGE			x	x	x			QUEUE	x	x	x	x	x	x	
COMPARE		x						RELEASE	x	x	x	x	x	x	x
COUNT				x	x	x		REMOVE				x	x	x	
DEPART			x	x	x	x	x	RETURN	x	x	x	x	x	x	x
ENTER	x	x	x	x	x	x	x	SAVAIL					x	x	
EXAMINE				x	x	x		SAVEVALUE			x	x	x	x	
EXECUTE			x	x	x	x		SAVEX		x					
FAVAIL					x	x		SCAN				x	x	x	
FUNAVAIL					x	x		SEIZE	x	x	x	x	x	x	x
GATE	x	x	x	x	x	x		SELECT				x	x	x	x
GATHER			x	x	x	x		SPLIT	x	x	x	x	x	x	x
GENERATE	x	x	x	x	x	x	x	STORE	x	x					
GOTO							x	SUNAVAIL					x	x	
HELP		x	x	x	x		x	TABULATE	x	x	x	x	x	x	x
HOLD	x	x						TAG	x						
IF							x	TERMINATE	x	x	x	x	x	x	x
INDEX		x	x	x	x	x		TEST			x	x	x	x	
INTERRUPT	x	x						TRACE		x	x	x	x	x	
JOIN				x	x	x		TRANSFER	x		x	x	x	x	
LEAVE	x		x	x	x	x	x	UNLINK			x	x	x	x	
LET							x	UNTRACE		x	x	x	x	x	
LINK			x	x	x	x		WAITIF							x
LOGIC		x	x	x	x	x		WRITE	x	x	x	x	x		

I = GPSS 1961, II = GPSS II, III = GPSS III, 6 = GPSS/360, V = GPSS V, P = GPSS/PC and μ = micro-GPSS

GPSS/H, Proof Animation and SLX are trademarks of Wolverine Software Corporation. GPSS/PC and GPSS World are trademarks of Minuteman Software Corporation.

REFERENCES

Abed, S., T. Barta and K. McRoberts. 1985. A quantitative comparison of three simulation languages: GPSS/H, SLAM, SIMSCRIPT. *Comput. Ind. Eng.*, 9, pp. 45-66.

Avens-Aventiņš J., J. Merkurjevs J. and J. Tolujevs. 1987. *Objektu un elastīgo ražotņu vadības algoritmu modeļi Metodiskie norādījumi imitācijas modelēšanā valodā GPSS.* Rīgas politehniskais institūts, Rīga,

Ball, D. 1992. GPSS/VI. In J. J. Swain, D. Goldsman, R. Crain and J. Wilson (eds.) *Proceedings of the 1992 Winter Simulation Conference*. La Jolla, CA.

Banks, J., J. Carson and J. Sy. 1989. *Getting Started with GPSS/H*. Wolverine Software, Annadale, VA. 1989

Behlau, T. and V. Hinz. GPSS*. 1993. A GPSS implementation with hierarchical modeling features. In G. Evans, M. Mollaghasemi, E. Russell and W. Biles (eds.) *Proceedings of the 1993 Winter Simulation Conference.* Los Angeles.

Behlau, T. and P. Lorenz. 1993. Building an even better GPSS. In *GPSS-Users' Group Europe-Gründungs-veranstaltung*. ASIM Heft 36, Magdeburg.

Beikirch, J. 1997. *Ein Java basiertes GPSS*. Diplomarbeit Otto-von-Guericke Universität, Magdeburg.

Bobillier, P., B. Kahan and A. Probst. 1976. *Simulation with GPSS and GPSS V.* Prentice Hall, Englewood Cliffs, NJ.

Born, R. 1998. Teaching Simulation of Manufacturing Systems Through Stepwise Refinement Using micro-GPSS. In D. Davani and D. Elizandro (eds.) *International Conference on Simulation and Multimedia in Engineering Education (ICSEE '98),* SCS, La Jolla.

Chisman, J.A. 1992. *Introduction to Simulation Modeling Using GPSS/PC*. Prentice Hall, Englewood Cliffs, N.J.

Christy, D. and H. Watson. 1983. The Application of Simulation: A Survey of Industry Practices. *Interfaces*, Vol. 13, No. 5.

Claeys, F., H. Vangheluwe and G. Vansteenkiste. 1995. HGPSS: A hierarchical extension to GPSS In M. Snorek, M. Sujansky and A. Verbraeck (eds.) *Proceedings of the 1995 European Simulation Multiconference.* Prague.

Crain, R. 1996. Simulation Using GPSS/H. In J. Charnes, D. Morrice, D.Brunner & J. Swain (eds.) *Proceedings of the 1996 Winter Simulation Conference*. Coronado, CA.

Diedenhofen, H. 1993. GRAMOS-GPSS. In *GPSS-Users' Group Europe Gründungs-veranstaltung*. ASIM Heft Nr. 36, Magdeburg.

Dietmeyer, D, G. Gordon, J. Runyon and B. Tague. 1960. (Conference paper CP-60-1090) An interpretative simulation program for estimating occupancy and delay in traffic-handling systems which are incompletely detailed. *IEEE Pacific general meeting*. San Diego.

Efron, R. and G. Gordon. 1964. A general purpose digital simulator and examples of its applications. *IBM Systems J.*, 3(1),

Frank, M. and P. Lorenz. 1979. *Simulation diskreter Prozesse*. Fachbuchverlag, Leipzig.

Gordon, G. 1961. *Preliminary manual for GPS – A general purpose systems simulator*. (Technical memorandum 17-048). IBM, White Plains, N.Y. October 6, 1961

Gordon, G. 1961b. A general purpose systems simulation program. In *Proc. EJCC*. Washington, D.C., pp. 87-104. Macmillan, NY.

Gordon, G. 1962. A general purpose systems simulator. *IBM Systems Journal*. 1, pp. 18 – 32.

Gordon, G. 1969. *System Simulation*. Prentice Hall, Englewood Cliffs, NJ. 1969 (2nd edition 1978)

Gordon, G. 1975. *The Application of GPSS to Discrete Simulation*. Prentice Hall. Englewood Cliffs, NJ.

Gordon, G. 1978. *The Development of the General Purpose Simulation System (GPSS).* In ACM SIGPLAN Notices, Vol. 13, No. 8, August. Also in R. Wexelblatt (Ed.) *History of Programming Languages*, Academic Press, NY, 1981.

Gordon, G. 1979. The design of the GPSS language. In Adams, R. and A. Dagramici (eds.) *Current Issues in Simulation*. Wiley, NY.

Gould, R. 1969. GPSS/360 – an improved general systems simulator. *IBM Systems Journal*, 8(1).

Greenberg, S. 1972. *GPSS Primer*. Wiley, NY.

Henriksen, J. 1983. *State-of-the-Art GPSS.* Paper presented at the 1983 Summer Computer Simulation Conference, Vancouver.

Henriksen, J. 1985. *The Development of GPSS/85*. Paper presented at the 18th Annual Simulation Symposium, Tampa, Florida.

Henriksen, J. 1993. *SLX, The Successor to GPSS/H.* In G. Evans, M. Mollaghasemi, E. Russell and W. Biles (eds.). *Proceedings of the 1993 Winter Simulation Conference.* Los Angeles, pp. 110-117.

Herper, H. and I. Ståhl. 1999. Micro-GPSS on the Web and for Windows: A tool for introduction to simulation in high schools. In P. Farrington, H. Nembhard, D. Sturrock and G. Evans (eds.) *Proceedings of the 1999 Winter Simulation Conference*. Phoenix, AZ.

Herscovitch, H. and T. Schneider. 1985. GPSS III - An extended general purpose simulator. *IBM Systems Journal*, 4(3).

IBM. 1963. *General purpose systems simulator II*. Form B20-6346. IBM Corp. White Plains, NY.

IBM. 1976 . *APL GPSS for APLSV* (program no. 5796-PJF).

IBM. 1981. *PL/I GPSS Users Manual.* IBM, White Plains, NY.

Karian, Z., and E. Dudewicz. 1991. *Modern Statistical. Systems, and GPSS Simulation*. Computer Science Press, NY.

Knepper, L and A. Krönchen. 1993. Überlegungen zu einer Simulationsumgebung von GPSS/H. In *GPSS-Users'*

Group Europe Gründungsveranstaltung, ASIM Heft nr. 36, Magdeburg.

Lorenz, P. and T. Schriber. 1996. Teaching introductory simulation. In J. Charnes, D. Morrice, D.Brunner & J. Swain (eds.) *Proceedings of the 1996 Winter Simulation Conference*. Coronado, CA.

Mathewson S. 1989. Simulation Support Environments. In Pidd, M. (Ed.) *Computer Modelling for Discrete Simulation*. Wiley, Chichester, UK.

McHaney, R. 1996. *Simulation Project Success and Failure: Some Survey Findings*. Working paper, Dept. of Management, Kansas State U., Manhattan.

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

Nance, R. 1996. A History of Discrete Simulation Programming Languages. In T. Bergin and R. Gibson (eds.). *History of Programming Languages II*. ACM Press, New York.

O'Donavan, T. 1979. *GPSS – Simulation Made Simple*. Wiley, Chichester.

O'Keefe, R. 1989. The Role of Artificial Intelligence in Discrete-Event Simulation. In Widman, L, K. Loparo and N. Nielsen. *Artificial Intelligence Simulation Modeling*. Wiley, NY.

Pawletta, T., W. Drewelow and S. Pawletta. 1998. Transaktionsorientierte Simulation in interaktiven SCEs. In Lorenz and B. Preim (Hrsg.) *Simulation und Visualisierung '98*. SCS, Magdeburg.

Preuss, F. 1987. *Forderungen an eine interaktive Nutzung von Simulationssystemen und ihre Realisierung durch Erweiterungen zum PS SIMDIS*. Dissertation, TU Magdeburg.

Reitman, J. 1971. *Computer Simulation Applications*. Wiley Interscience, New York.

Richards, B. 1987. Introduction to GPSS. In J. Caroll *Simulation Using Personal Computers.* Prentice Hall, Englewood Cliffs, NJ.

Richards, B. 2001. *Email correspondence* with the author.

Rodrigues, J. 1993. *Directory of Simulation Software*. SCS, p. 23

Roesmann, H. 1979. *Simulation mit GPSS*. Oldenbourg Verlag, München.

Schmidt, B. 1979. *GPSS-FORTRAN.* Springer, Berlin.

Schriber, T. 1974. *Simulation Using GPSS*. Wiley, New York.

Schriber, T. 1991. *An Introduction to Simulation Using GPSS/H*. Wiley, NY. (Russian-language edition, Mashinostroyenie Press, Moscow, 1980).

Schriber, T. and D. Brunner. 1998. How discrete-event simulation software works. In J. Banks (ed.) *Handbook of Simulation*. Wiley, NY.

Schulze, T. 1991. *GPSS–Implementationen auf Personalcomputern dargestellt am Beispiels des Systems SIMPC*. Habilitation, TU Magdeburg.

Schulze, T. and J. Henriksen. 1998. *Simulation Needs SLX*. Otto-von-Guericke Universität, Magdeburg.

Seprödi, L. 1980. A GPSS Szimulációs Nyelv, Müszaki Könyvkiadó, Budapest.

Silverman, R. 1997. *A Laboratory Manual for Simulation with GPSS/H for Computer Science Majors*. University of the District of Columbia.

Solomon, S. 1983. *Simulation of Waiting-Line Systems*. Prentice Hall, Englewood Cliffs, NJ.

Stefanov, S. 1975. *Simulatsionna Modelinare chez GPSS/360*. Karl Marx Institute, Sofia.

Steinhausen, D. 1994. *Simulationstechniken*. Oldenbourg Verlag, München.

Ståhl, I. 1990. *Introduction to Simulation with GPSS: On the PC, Macintosh and VAX.* Prentice Hall International, Hemel Hempstead, U.K. 1990

Ståhl, I. 1993. GPSS will prevail - Some reasons for the resilience of the GPSS simulation ideas. In *GPSS-Users' Group Europe–Gründungsveranstaltung.* ASIM Heft Nr. 36, Magdeburg.

Ståhl, I. 1996. *Simulation Made Simple Using micro-GPSS.* SSE, Stockholm. 1996.

Ståhl, I. 1998.The Efficient Implementation of Wait Statements. In D. J. Medeiros, E.F. Watson, J. S. Carson and S. Manivannan (eds.) *1998 Winter Simulation Conference Proceedings*. Washington, D.C.

Ståhl, I. and F. Hall. 1999. Principles for a Web-based Micro-GPSS System for Swedish High-School Students. In H. Szczerbicka (eds.) *Modelling and Simulation, A Tool for the next Millenium.* SCS, Warsawa. 1999.

Ståhl, I. and H. Herper. 1998. *Einführung in die Simulation mit Micro-GPSS - Ein Grundkurs in acht Lektionen.* Universität Magdeburg.

Thesen, A. and L. Travis. 1992. *Simulation for Decision Making*. West Publishing, St. Paul, MN. 1992

Tocher, K. 1966. Some techniques of model building. *Proc. IBM Scientific Computing Symposium on Simulation Models and Gaming*. IBM, White Plains, NY.

Tocher, K. and D. Owen. 1960. The Automatic Programming of Simulations. In *Proc. of the Third International Conference on Operational Research*.

Yackiv, I., J. Tolujev and N. Sinenko. 1994. Sistema Imitatsionnavo Modelirovanija GPSS/PC. Riga Aviation University.

Uyeno, D. and W. Vaessen. 1980. PASSIM: a discrete-event simulation package for Pascal. *Simulation*, 35, p. 183-190.

Vaucher, J. 1977. <http://www.jsp.umontreal.ca/~vaucher/Software/gpsss.sim>.

Watson, H. and J. Blackstone. 1989. *Computer Simulation.* 2nd Ed. Wiley, NY.

Weber, K., R. Trzebiner and H. Tempelmeier. 1983. *Simulation mit GPSS*. Stuttgart.

Wolverine Software 1993. *Checkpoint*, Fall 1993, p. 7.