Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

A HISTORY OF SIMULATION DEVELOPMENT IN THE UNITED KINGDOM

Brian W Hollocks

Faculty of Management
Bournemouth University
Executive Business Centre, Holdenhurst Road
Bournemouth BH8 8EB, UNITED KINGDOM

ABSTRACT

Discrete-event simulation first emerged in the late 1950s and steadily grew in popularity to become the most frequently used of the classical Operational Research techniques across a range of industries and users. The leading advances in the evolution of discrete-event simulation software came from the United Kingdom and the USA and the author was engaged for some 30 years with its development and use. The paper reviews that history as a first-hand account, specifically in the United Kingdom and focusing on the period to 1994.

1 INTRODUCTION AND CONTEXT

This paper describes the emergence of discrete-event simulation software in the United Kingdom (UK). It is based on the personal experience and first-hand observations of the author and addresses primarily the early years, taken here to be up to 1994. Other personal reflections can be found in earlier papers by the author (Hollocks 2006a, 2006b, 2008).

I came to simulation in 1964 by way of Operational Research (OR), which I had in turn been introduced to in the final year of my Engineering studies. I was attracted by the idea of the use of representative models to explore real-world problems. At that time, OR was still something of a minority activity, but growing, and most major companies were investing in it across a diverse range of industries. For example, the UK's National Coal Board (NCB) had initiated OR around 1948 (although at first under the title of Field Investigation Group), following the steel industry, where the British Iron & Steel Association (BISRA) enlisted its first OR employee in 1945. BISRA published a "The First 20 Years" report in 1965 (Collcutt 1965). By 1964 numerous industrial OR groups were emerging outside of "big" industry, but the problem for me in entering OR was the emphasis placed by many groups on recruiting only mathematicians.

The Operational Research Club (sic) had been founded in the UK in 1948, produced a quarterly journal from 1950, and became the Operational Research Society in 1953. By 1964 membership had reached 1,242 (Cummings 2008). However, by 1964 there were OR programs in only two UK universities, Imperial College, London and Birmingham University. The first university OR Department as such was opened in 1964 at the then new University of Lancaster.

The group that I eventually joined was the United Steel Companies' central OR department in Sheffield, Yorkshire. This had been originally founded by Stafford Beer whose principal personal interest was Cybernetics. In 1956 he had persuaded the Board of United Steels to establish a corporate "Department of OR and Cybernetics". The following year a large detached house was acquired in an attractive suburb of Sheffield as the base for this new department and it was renamed "Cybor" House (Hollocks 2006a). For it, Beer sought a high-quality multi-disciplinary team (disciplines which ranged from Zoology and Biology through English and Economics to Mathematics and Statistics). In 1958 the adjacent property to the Cybor House building (known as Redlands) was added. Not only did this provide added staff accommodation to a growing department but it facilitated acquisition of a Ferranti Pegasus Mk II computer (shared with Sheffield University). Figure 1 shows Cybor House to the left and Redlands to the right (including a late 1960s extension). It illustrates the nature of the investment that Beer secured. Co-incidentally, BISRA had already acquired a Ferranti Pegasus at its base in Battersea, London.

Figure 1: Cybor House and Redlands.

The "Cybernetics" of the United Steel department's title was little present as such in the actual day to day work and in 1961, when Stafford Beer left the company, the emphasis on OR was consolidated. Outside of OR there had been some particular activity in Process Control, such as in data logging for which the department had a purpose-built device (the Unisteel Automatic Recorder, UAR) and a further computer, a Ferranti Argus 500. Equipment was mounted in a truck such that it could be taken to steelworks for projects. By 1964 the Department was led by Dr KD Tocher (Figure 2), commonly referred to as "Toch", who had been recruited from Imperial College around 1957, and the staffing of the Department had reached 80-90.

Figure 2: Prof KD Tocher (1921-81).

2 ENTER SIMULATION

Through the 1950s and 1960s OR generally had been assembling an arsenal of tools, principally drawing on mathematics and statistics, and text books were emerging such as Sasieni, Yaspan & Friedman (1959). Along with the likes of inventory control, mathematical programming, and game theory, the tools commonly included queueing problems with Monte Carlo methods to address the statistics. Tocher had already developed an interest in this field well before arriving at Cybor House. At a meeting of the Operational Research Club in 1952 on “Marshalling and Queuing”, Tocher pointed out in his paper on "Some Unsolved Problems" "that the new electronic automatic computing machines could be harnessed to solve all such problems by reducing them to an abstract system" (Tocher 1952). Interestingly he went on to say that he "did not feel that the use of these "Monte Carlo" methods should be regarded as anything but a stop-gap procedure"! (I have discussed Tocher's direct contribution to the development of simulation elsewhere (Hollocks 2008).) To assist position this era in computing terms, the first UK business computer, the LEO I at J. Lyons and Company, went into operation in November 1951 (Mowery 2003).

However, programs for the "automatic computing machines" could only be written in the intricate machine-level codes that were available for them. The limitations of using machine code for the production of any complex programs are clear, including slow to write and difficult to debug, and such problems were the driving force for the development elsewhere of high-level languages. The impact of using low-level facilities for simulation is illustrated by the development of a simulation of steelmaking operations in the Steel Company of Wales that Neate and Dacey reported (1958) as consuming two man years of work in designing and coding a model (on the BISRA Ferranti Pegasus) - and which had still to be tested!

This was not the first UK computational representation of a system however. In the 1950's, some research institutions, for example, at the Post Office Research Station at Dollis Hill, London, quoted by Tocher at the 1952 Operational Research Club meeting referred to above, constructed special devices to represent particular stochastic queuing systems, such as for telephone traffic. A mechanical "analogue randomizer" was designed by Stafford Beer at a United Steel Companies’ steel-plant, Samuel Fox & Co, and ten built to study the behavior of complex queues (Tocher 1963). I only saw the remains of one and have not yet found any record, or even anecdote, of their actual application.

In the relatively early days of his time with United Steel Companies, Tocher was faced with constructing a simulation model of one of the United Steel Companies' steelplants (Hollocks 2006a). Since United Steels had several steel plants (with differing technologies - Open Hearth and Electric Arc at the time), he envisaged that a standard model could be created such that parameter changes would permit it to represent any of the plants. In seeking to generalize the model to accommodate the various layouts, equipment configurations, operating rules and processes across the company, the concept moved from a *general steelplant program* to a *General Simulation Program* (GSP). His basic framework for the

Figure 3: Ferranti Pegasus II at Cybor House.

construction of simulation models was conceived in 1957 and was hence the first identifiable specialist package (first published in Tocher and Owen 1960). Interestingly, the seminal Tocher and Owen (1960) paper also links the simulation work to the Cybor House data logging activity referred to earlier - as a means of securing adequate data.

The GSP software was developed on the Ferranti Pegasus II, a valve-based first-generation computer (shown in Figure 3, before the magnetic tape drives were installed along the back wall of the computer room). The GSP work is particularly notable for the fact that it pre-dated the availability of high-level languages. Fortran did not emerge in the USA until 1957-1958, let alone become available widely, nor Algol in Europe until 1960 (Backus and Nauer, 1960). GSP started with a Mark 0 but only one project utilized that version prior to the availability of Mark I (Hollocks (2006) quoting one of Tocher's colleagues, Peter Amiry).

It is of interest that GSP's fundamental structure was informed/inspired by the needs of steel plant modelling, where the dominant element is the furnaces with their cyclical and batch nature. This conceivably also colored the general world-view of systems as sets of "machines" changing "state" at "events", leading Tocher to his iconic three-phase structure (Tocher and Owen 1960). He mapped this concept onto the central Pegasus computing-store architecture which included 8 accumulators, facilitating his creating a central data structure of an 8-column matrix (the columns identified as S to Z) each row of which could potentially represent a "machine".

By this time, Gordon had produced GPSS at IBM (Gordon 1962). His design, in turn, was informed by *its* context - the goal of modelling designs and problems within computer systems. GPSS was made available by IBM as bundled software with its computers. The essence of Gordon's view was of the flow of some entity through a system, eg data. More generally this could be regarded as the flow of "material" through a physical system. The issue of the (GSP) "machine" view of the world versus the (GPSS) "material" view of the world was a common philosophical debate in simulation circles for many years.

In the UK, BISRA, based in London, had produced a software tool, Montecode (Collcutt 1965, Head 1962), to support its use of Monte Carlo methods and there were other tools, developed within universities or by computer hardware suppliers. These typically took the form of a library of routines for use with other software (for example the Elliott Simulation Package, ESP (Williams 1962)).

By 1964 the General Simulation Program had progressed to a Mk II (Tocher and Hopkins 1964) and that package was to be *my* first exposure to both simulation and GSP. The design had a distinctive syntax but was not difficult to learn (Figure 4 shows an example of the code). Numerous simulations were undertaken with it and its principles proved sound. It was, however, still operating on the Pegasus II, a 5-hole punched-tape fed computer that filled a good sized room – complete with its multiple tape-decks, magnetic drum memory and teleprinter output. Its internal memory (RAM) was 64 words of 39 bits! (Run speed was quite slow; I recall evenings (or nights!) spent watching over the machine as it ran simulations at 30 minutes real time per simulated week.)

Within a year, the department was taking delivery of a new machine – an Elliott Automation 503 (Figure 5). This was transistor-based with greater internal memory, and for wider use came with the Algol60 programming language (Backus and Naur 1960). At first the 503 was linked to the Pegasus and its tape drives, but later acquired its own storage device, a distinctive cartridge system from a US company called Potter involving continuous loops of tape on air bearings. GSP was then ported to the 503, and rewritten to a Mark III. The Pegasus was then consigned to scrap.

A practical drawback to the GSP work of Tocher and his team was that it was written in machine-code for a specific scientific computer (and with specific hardware modifications designed and built in a lab at Cybor House) and was therefore, in effect, not portable. In contrast to this, the emergence of the Fortran and Algol high-level languages was beginning to permit portability of software between computer installations, and even between makes of computer, albeit with some modifications in the detailed format for the high-level language as implemented for that manufacturer's equipment (such as in input-output features).

```
C6(START TRANSPORT FROM M.SHOP)

L5,K60
J1,W>0
FIND(0,W9,X9)
Kn,S=0.
v1→MIN(U,Z17,W60)
n2→V,v2→v1
*n1→0
L2,X(n2+n1)→W(61+n1)
DIS(TIM-W(W(61+n1)),5)
n1+1
v2-1
J2,v2*>0
MIST(60,4,0,v1)
K105,v→NOR(6,X,Y)
v+NEG(6,Z)
Kn,Z→TIM
S→1
T→TIM+v
Y+v
W→v1
v1→T
J5,K9,T>v1,S→n
D(v1-TIM),J5.
L1,CLOSE(6).

C7(SMALL ARC PIT STARTS STRIP)

n1→5
Kn1,X→T(V)
T(V)→1
J1,K12,U>0
U-1
K5
STRIP
S→3
CLOSE(7).
L1,Z13+1.
```

Figure 4: Example of GSP II code.

Figure 5: Console of Cybor House Elliott 503, installed 1964-5.

Nb: It has been suggested that the first contact between UK development through Tocher and US developments, as represented by Gordon, Kiviat and others, was an IBM symposium on Simulation and Gaming in Washington in 1965. In due course, Tocher gave the Keynote Address at the 1979 Winter Simulation Conference (Tocher 1979).

By this point in time (1965) I had transferred from the Central OR group to one of the United Steel's Works (Samuel Fox & Co.) where I found that they had acquired a computer for data-processing applications, namely an IBM 1460 with 80-column punched card feed and a line-printer - and Fortran IV. With this available, I set about creating a framework of routines which could support constructing simulation models. (Later conversations with colleagues in the wider OR community indicated that producing such an in-house toolkit was, at the time, a not-uncommon initiative amongst OR staff!) My activity drew on what had just emerged as the first simulation textbook (Naylor et al, 1966), as well as Tocher's seminal book (Tocher 1963). (Tocher's book was the first published on simulation but he had the misfortune of his original typed text being lost in a fire at the publishers and having to rewrite it!)

The downside of the Fortran-based simulation facility was the difficulty in obtaining service from the Data Processing Department, as it was by then called. Standard service (for data-preparation or runs) was an over-night turn-round; a concept normal at the time, but difficult to envisage today! Together with the limited debugging/development facilities that could be engineered in such a subprogram-library based tool, simulation model development was a slow business.

In the UK more widely, further simulation software was produced based on the principles established by Tocher but using high-level languages. John Buxton, who had worked at BISRA on Montecode, and John Laski, who had worked on GSP Mark I, conceived the Control and Simulation Language – CSL (Buxton and Laski 1969) - within Esso Petroleum in a joint project with IBM and using Fortran. Hills (1965) produced SIMON, based on Algol, as the subject of his MSc thesis at what was then Bristol College of Science & Technology (now University of Bath). These products, together with GPSS, became commonly available in computer manufacturers' software catalogues, this being the era of 'bundled' software (that is, supplied free with hardware).

As an aside, it is interesting to note that, in the early days, manual simulation was a not-uncommon practice: it was simple, required no sophisticated facilities, and a model could be conceived and initiated virtually immediately. However, the major drawbacks were rapidly evident. Even modest-sized problems are extremely time-consuming to process, and the method is prone to human error since the logic of operations which must be followed in order to execute even a simple practical simulation can be intricate enough to permit mistakes and misunderstandings. Since the purpose of simulation is experimentation, repeat runs are essential and the time required for manual simulation of even small systems is prohibitive. Formal manual methods were developed in some organizations, for example the coal industry (Szabo and Lyons 1971), but the best known manual format in the UK was that of HOCUS - Hand Or Computer Universal Simulator (Hills and Poole 1969) from PE Consultants. This also, or perhaps principally, acted as a pre-processor prior to encoding the manual form as input to a computer program.

Beyond the UK, activity in simulation software elsewhere in Europe was more limited. A significant development was in the Norwegian Computer Centre who, partially funded under a research contract with Univac, produced in the mid-60s a language for the dual purpose of system description and simulation programming - SIMULA (Dahl and Nygaard 1965). This was a super-set of Algol-60 and has subsequently been seen as more significant for its pioneering of object-orientation than as a simulation package.

After UK steel nationalization in 1967 (creating the British Steel Corporation, BSC), I returned to Cybor House in 1968 and to the use of GSP III on the Elliott 503. GSP, which by then incorporated a list processing suite, was a powerful way of representing systems, but the 503 computer was ageing and new hardware was being acquired for the site – in particular an ICL 1900 series system. (Machines more suited to scientific work were becoming generally scarcer.) The company had amassed a varied range of equipment and suppliers through the nationalization and decided, inevitably perhaps, to standardize on two suppliers for all its mainframes – IBM and ICL. As the Cybor House 503 became older and more unreliable, another source of simulation processing was required and it was highly desirable that it was compatible with GSP. In 1970, with the support of the department, I re-implemented GSP III's functions and features using Fortran as a platform (referring to the result rather unimaginatively as FORSS - Fortran-based Simulation System), using the ICL 1904E machine acquired by the Systems group then also occupying Cybor House. With the intrinsic portability of Fortran, this GSP version could potentially be implemented on a range of common computer hardware.

In 1971 I moved from Cybor House to the OR group at British Steel's division on Teesside in the North East of England, accompanied by a copy of FORSS (on 2000 punched cards in a cardboard box). The Teesside OR team had previously been using GPSS, being a long-standing IBM site. Porting FORSS from ICL to IBM proved no problem and through the 1970s it was further refined at Teesside. It progressively became more widely used across much of the Corporation (British Steel 1975).

Staff in (the solely IBM based) South Wales sites of British Steel produced what amounted to a PL/1 equivalent, although with more divergence from GSP's specification. Tocher and his team back at Cybor House produced by the mid-1970s a pilot version of GSP Mark IV (Bent 1976). This was part of an ambitious wider project that, firstly, generalized the syntax of GSP into a Language for OR in British Steel (LORBS), and, secondly, constructed the application in a purpose-designed Machine Independent Low-level Language (MILL). The goal, as the MILL name indicates, was to permit the implementation of GSP or other application on any computer system by creating for it a focused MILL compiler (which was quite small). However, the main MILL application, taking a considerable amount of OR attention at that time, was not simulation but a tool for creating planning models: a Language for Economic Modelling in British Steel (LEMBS). Although the general architecture worked (albeit with slow compile times), GSP IV failed to gather much traction - even though Toch himself moved from Cybor House to an influential British Steel Head Office advisory role (based in Birmingham). GSP IV may have suffered from being in the shadow of LEMBS or, by then, from the momentum of FORSS amongst users.

Despite all of this activity in the UK and Europe (not to mention the USA), Christy & Watson (1983) found, in a 1980s survey of industrial use of simulation, that 80% of models were still written in Fortran!

3 COMPUTING RESOURCES

Through the 1970s, existing commercial simulation software continued to be ported to new hardware as it became available, and new versions were produced. The emergence of mini-computers (such as from Digital Equipment Corporation (DEC) and Data General) through this period made little impact in simulation. However, the later availability of *micro*-computers, and then personal computers, stimulated interest more and facilitated the next major shift in simulation.

A major attraction of the microcomputer to the OR community and other computing users was the by-passing of the Data Processing department bottleneck in computer-power access. The micro offered direct access to a power useable for practical tasks. It is difficult to appreciate now just how exciting the advent of the desk-top computers, was.

Although the Apple II (with, in those days, an open architecture but still a distinctive operating system) was the flagship image of the micro revolution, the employment of the micro in simulation relied more on the up-market 8-bit/16-bit machines, for example from builders such as Altos and Cromemco. As an illustration, Alan Clementson, based at Birmingham University, used special program management techniques to shoehorn his version of CSL into an Altos micro-computer (Clementson 1981). However, John Crookes, for example, at Lancaster University (Crookes and Valentine 1982, Crookes 1983) was an enthusiastic advocate of the lower cost and widely available Apple II and produced models operating on that equipment, including a simulation system based on Tocher's 3-phase structure (Crookes et al 1986). This linked with other work on a Computer Aided Simulation Modelling (CAPM) framework being carried out at London School of Economics under Ray Paul and David Balmer (Balmer and Paul 1986).

The use of the micro-computer became a favorite conference/seminar topic of the period in the UK OR community, for example Lines (1981) and Ranyard (1981) - indeed the OR Society had a Micro-computer

Study Group for a time. The whole technological development acquired respectability when IBM released its first Personal Computer (model number 5150) in 1981.

(Just before this period (in 1977) I had returned once again to Cybor House – by then being the "English OR Unit" of British Steel. The group still had software development resources, something that we would exploit significantly in the near future.)

4 THE IMPACT OF GRAPHICS

In one of the very early (late-1950s) projects using GSP Mark I, Tocher's team were required to provide some visual means by which a real-world decision maker could understand what was going on in the simulation model of a steel melting shop. The team did this by producing a physical representation of that melting shop on a large display board, some 6 feet by 3 feet, which, together with the simulated time as it progressed, was updated from output from the computer simulation model as it was running (Tocher 1960).

Given a dedicated computer available (the Ferranti Pegasus first and the Elliott 503 later), Tocher's team were also able to use such displays electronically connected to the computer running a simulation. Hence controlling the model from the console alongside the display provided interactions with the running simulation in such a way that decisions could be made externally to the model by a decision-maker given the simulated circumstances. This approach of so-called "production games" was used, for example, in developing scheduling rules for furnaces and rolling mills in the early-1960s (Mellor and Tocher 1963). A GSP-driven game activity is illustrated in Figure 6. The ideas of display and interaction were powerful ones but impractical to use on a routine basis. Displays were either crude, or expensive to create.

In the early/mid-1970's Bob Hurrion, as part of his PhD research in scheduling at Imperial College, London, tested the use of mimic diagrams with simulation (Hurrion 1976). This work was evidently based on a version of SIMON (Mathewson 1977) and, initially, used alphanumeric visual display facilities. However, Hurrion refined the principles and his implementation with a low cost color-graphics terminal which enabled dynamic color mimic diagrams to be driven from a simulation model (Hurrion 1981). Hurrion's prototype software not only accommodated graphics but also the ability to interact with certain parameters of the simulation.

Figure 6: 1960s Production "Game" in action.

Hurrion's work generated considerable interest in the OR community. A subsidiary of then UK motor manufacturer British Leyland (BL Systems) was at the time, 1978-9, seeking a tool to improve their simulation capability. They were faced with a need to simulate plant development plans (Fiddy et al 1981) for a new car critical to the Company's success. The initiative with Bob Hurrion (by then at the University of Warwick) followed manual operation of a visual presentation from output of a model written in GPSS (Figure 7).

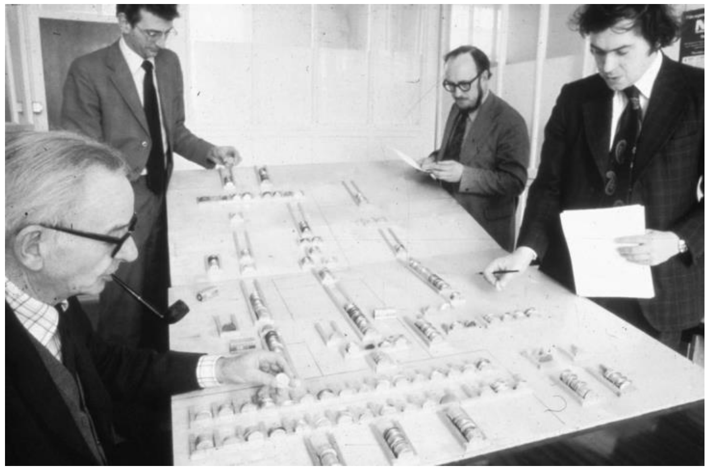

Figure 7: BL Systems Manual Display based on GPSS Output Data.

. Hurrion's graphics animation proved so valuable to BL Systems that they subsequently arranged to re-code and market the product commercially using the name SEE WHY (BL Systems 1980). The product was based on Fortran and initially launched on DEC-type mini-computers linked to an Intecolor programmable color terminal. The implementation was soon moved to a micro-computer from Cromemco - originally 8-bit but later 16-bit using a Unix-type operating system (a configuration is shown in Figure 8). The attractiveness of graphics/mimic diagrams lay not only in their potential to demonstrate a model to a decision maker, but also to assist model builders to more easily or thoroughly test their models.

The interest in Hurrion's work extended also to the British Steel OR community, who noted the potential of mimic diagrams. Since they were unable to acquire separately the graphics of SEE WHY (SEE WHY being an integrated whole) to add to FORSS, in which there was considerable expertise invested and many legacy models in the company, work was initiated using the Cybor House software team to extend the FORSS package to have animated graphics facilities. The initial version of this was completed in 1980 and christened FORSSIGHT (Hollocks 1982). This was initially based on the ICL and IBM mainframes then common across the British Steel Corporation (BSC), also using an Intecolor terminal (Figure 9) – there

Figure 8: Typical Cromemco-based SEE WHY System.

being no alternative as practical. It also generated interest from outside of BSC and, after being unsuccessful in establishing an agent to sell the product, the team were drawn, reluctantly, into marketing it themselves (trading under the name Business Science Computing).

Figure 9: Early FORSSIGHT System.

In 1980, the English OR Unit finally moved from Cybor House (which was sold) to another site in Sheffield - initially, to offices once used by part of BISRA. In a material sense, it was the end of an era.

In this period, FORSSIGHT was ported to 16-bit micro-computers, using the p-System operating environment (mostly on the Sage micro). With this move, the commercial business started to take on a real life of its own. (This was a quite distinct activity from the wider OR being carried out for British Steel clients by the remainder of the group.) It was a fascinating time. Simulation users overseas showed an interest and eventually a US base was established. To this end, FORSSIGHT had to be re-badged for the US market to avoid confusion with an established forecasting package. After some failed attempts at a new name, "Witness" was proposed and proved acceptable. There were other developments in the UK, for example PE Consultants progressed HOCUS to incorporate graphics, and more US products were marketed, eg SLAM/TESS, SIMAN/Cinema.

Although the major (and varied) developments in simulation in the US are not part of the brief of this paper, it is perhaps worth observing a divergence between the US and UK development paths in adding graphics. The mainstream UK thread, for example SEE WHY/FORSSIGHT, integrated the graphics display with the simulation model such that the graphics display was updated by the model as it ran, ie the graphics are "concurrent" with the model. Hence to stop the model is to stop the graphics and vice versa. The principal USA development path in graphics, I observed, at the time adopted a "replay" approach. In this, a simulation run produces an output data file detailing the sequence of events in the run; this is then the source for a second program which generates graphics from the file. Hence such graphics are only interactive *within* the graphics, for example pan and zoom or slow motion. There is no interaction with the model itself. The merits of the replay approach are the relative speed and smoothness of visual operation (since no event-processing is present). It can also, in principle, run backwards - not possible with concurrent graphics since no known discrete-event simulation software executive included event scheduling which could reverse. The continuing increase in the speed of computer processors led later simulation software developments to the concurrent approach.

In 1983 the BL Systems' Director responsible for the development of SEE WHY, and subsequently initiating commercial marketing, Ed Fiddy, resigned from the Company and set up his own business (Insight International, later Insight Logistics) near Oxford. This developed and marketed a range of Visual Interactive Modelling software, including graphics and simulation, initially under the general title OPTIK (Insight International, 1984). In 1984 BL Systems itself was rebranded ISTEL (the component letters having no acronym significance; the name was evidently chosen from 3 million generated by a computer program!). They subsequently approached the British Steel Corporation to acquire FORSSIGHT and the

Business Science Computing team. As part of its downsizing, BSC accepted and the team thus became ISTEL employees in 1984 and relocated.

The advent in 1984-5 of the IBM PC-AT with its 16-bit Intel processor and Microsoft Operating System (DOS) provided a desktop facility with adequate power to drive the more power-hungry of the micro-based simulations. This development was accompanied by the required graphics resolution to correspond to the quality existing in the free-standing graphics terminals used up to that time. Initially this used third party graphics management software but the IBM EGA (Enhanced Graphics Adapter) graphics standard became the convention when it became available. A higher resolution graphics standard, the PGA (Professional Graphics Adapter) was also announced at that time, but provided more than was typically required, although some simulation software made use of it.

The entry of IBM into the personal computer field had brought the PC corporate respectability. All major simulation software products ported to this hardware. The packages were the same, in principle, as had previously operated in the Cromemco, Sage and other environments. Later packages emerging in the UK market followed the same pattern, for example Taylor II (King 1996), from the Netherlands..

In this environment, simulation was an increasingly high-profile tool. It was a common discussion topic amongst OR professionals leading to a Special Interest Group in the OR Society.

5 FRONT ENDS AND SIMULATORS

A time consuming feature of simulation modelling is designing, writing and de-bugging a model's code. From the earliest days of simulation there had been interest in creating the means to make this more rapid and more reliable. Initially this led to the creation of simulation languages, as discussed earlier, as a replacement for writing in low level computer code. This was followed, as a natural continuation, by simulation program generators (Mathewson 1974), a "front-end" to languages.

An early pioneer in the UK was Clementson (1973) with CAPS (Computer Assisted Production of Simulations). This permitted the non-coding creation of simple simulation models through a system of menus describing the activities making up the simulated system. Any complex production rules or other sophistication had to be introduced through the coding of sub-programs which were linked into this interactively-created framework. At Imperial College Mathewson (1984) developed DRAFT, a program generator which could be used with more than one simulation package - initially addressing SIMON. Just prior to its acquisition by ISTEL, the British Steel/Business Science Computing team had designed and developed a front end for FORSSIGHT termed FORGE (FORssight GEnerator) but its life was overtaken by events. A different approach had been adopted by ISTEL who created in EXPRESS a front end to SEE WHY which was not a code generator as much as a simulation development environment to assist in prompting the creation and the editing of code (Shanehchi 1985). Front-ends continued to emerge for some time, for example ISI (Nolan et al 1991) developed in Ireland, which generated SIMAN or SLAM code.

These front-ends were a means of creating code for separate simulation packages or languages. A natural progression of such front-ends was to seek to avoid the creation of code entirely, in essence being a general model through the parameters of which the actual subject configuration could be represented. The objective was, as with front-ends, to avoid coding, but these general models aimed also to reduce the delays caused by compilation and linking. Averill Law and Jerry Banks later distinguished these tools with the term "Simulators", as distinct from simulation *languages* (Law and Haider 1989, Banks et al 1991). Simulators of a kind, ie specific generic models within tightly defined domains, had been created within simulation-using companies some time earlier - for example, a general mine model in the National Coal Board and a general plate mill model in BSC (British Steel, 1978).

ISTEL having acquired FORSSIGHT, consolidated its development/support teams and focused product development on an interactive form-driven simulator (written, in the first version, in SEE WHY) under the design leadership of Martin Clark (1988). This adopted the title previously secured for FORSSIGHT in the USA, namely WITNESS (Gilman and Watramez 1986). It caught on and a (much) later form is still active in the field (https://www.lanner.com/technology/witness-simulation-software.html).

An aim was to include more "intelligence" in a simulation package, that is, to enable it to carry out more activity on the part of the simulation user. Supporting this, it was desirable not only to eliminate coding, with the associated creation, entry and de-bugging, but also to adopt an interpretive mode of operation - comparable, for example, to that of BASIC (Kemeny and Kurtz 1964). Then a part or whole of a simulation model could be run immediately on creation, without compilation and linking; so, not only could parameters be changed within a simulation model, but the very components making up the model could be changed at any point also. This had significant potential for simulation methodology, as a simple model might be constructed and interactively evolved or refined in relevant areas as the model builder or user was satisfied (or otherwise) with the construction to date. The idea of simulators has shades of Tocher's original aspirations for a general simulation, as described earlier, but he rejected the idea of interpreted operation because of its impact, then, on run speed (Tocher and Owen 1960). As it transpired, simulators proved a more successful initiative than front-ends.

By the late 1980s, USA trade journal Managing Automation, quoting Survey company Dataquest, reported that Simulation was the fastest growing segment of the PC software market (Klein 1988).

The number of US products marketed in the UK was increasing, eg ProModel, Extend, MicroSaint. In the early 1990s a German-originated product known as SIMPLE++ (Becker 1994) became available in the UK, initially based on UNIX. It pursued an overt adoption of object-orientation, by this time becoming more popular in mainstream computing, and which had originally been conceived in the mid-1960s, in conjunction with simulation, as a feature of SIMULA as discussed earlier.

One of Ed Fiddy's team with Insight Logistics, Mark Elder, (who had been part of the original team working on SEE WHY) in his turn moved on in 1994 to Strathclyde University, developing his vision of a simulation product which he named SIMUL8 (Elder 1995, https://www.simul8.com). A management buy-out acquired ISTEL from its parent company (by then, the Rover Group) in 1987, and it was sold on, some 2-3 years later, to AT&T. In 1996 the simulation team, in turn, was the subject of a management buy-out from AT&T forming Lanner.

(Nb: Tocher left British Steel to become Professor of Operational Research at the University of Southampton in 1980 but died in 1981. Bob Hurrion died in 2014.)

6 CONCLUSION

The picture, looking back on the UK history reviewed above, is one driven by a mixture of individuals (with their interests and inspirations), enabling technologies, and industrial/business need or opportunity. Ease of model building dominated simulation software development as the driving force - first leading to structured packages, later to front ends, and subsequently to simulators. The progress in visual displays delivering animated computer graphics - superimposed on that evolution, upping the power and communication (and the marketability) of simulation. Robinson (2005) discusses further dimensions to simulation development beyond this phase, such as its integration with virtual reality tools.

Being a computationally intensive technology, simulation has continued to take advantage of wider improvements in computing. Reviews of software packages later in the hardware/software evolution, such as Pidd (1988) and Van Breedam et al (1990), gave no prominence to run-time as a key feature of choice between software but in the early days of simulation it was a frequent point of discussion (for example Laski 1965).

A further aspect of simulation activity has been the changing user population. The presence in organizations of large and distinct management science resources has reduced materially in the UK (a shift clearly identified in a mid-1990s UK study of OR groups (Fildes and Ranyard 1997)). The UK OR Society still has an active Simulation Special Interest Group and since December 2006 has published a Journal of Simulation.

I miss involvement now with the software development business, but the pioneering period up to 1994 was interesting and stimulating. Not that evolution is over for simulation; its need remains (indeed, there *still* seems scope for wider use) and technology may bring surprises. There is no evidence to suggest that the 92% satisfaction rate identified in the 1991 UK user survey (Hollocks 1992) has diminished.

ACKNOWLEDGMENTS

All trademarks referred to in this paper are acknowledged as appropriate. Thanks are due to those organizations that granted me leave to freely make use of the photographs included. (The others are my own.)

If readers are aware of material omissions from or errors in what I have described, please let me know. I have sought only to record what I found, observed or experienced first-hand. History continues.

REFERENCES

Backus, J. and P. Naur. 1960. "Report on the Algorithmic Language ALGOL 60". *Numerische Mathematik* **2**: 106-136.

Balmer, D.W. and Paul, R.J. 1986. "CASM - The Right Environment for Simulation". *Journal of the Operational Research Society.* **37**, 443-452.

Banks, J., E. Aviles, J.R. McLaughlin, and R.C. Yuan. 1991. "The Simulator: New member of the Simulation Family". *Interfaces* **21**: 76-86.

Becker, R. 1994. *SIMPLE++ - A New Class of Simulation Software.* Aesop GmbH, Stuttgart.

Bent, M.E. 1976. *Pilot GSP IV Manual*. British Steel Corporation Central Management Services, Sheffield. October.

BL Systems. 1980. *SEE WHY user manual*. BL Systems Limited, Redditch, UK.

British Steel. 1975. *FORSS - A Simulation System Based on Fortran.* Report OR67/1, British Steel Corporation, Teesside and Workington Group, Redcar .

British Steel. 1978. *Generalised plate mill modelling system*. Report OR 113/1, British Steel Corporation, Teesside Division, Middlesbrough.

Buxton, J.M. and J.G. Laski. 1969. "Control and Simulation Language". *The Computer Journal* **5**: 194-199.

Christy, D.P. and H.J. Watson. 1983. "The Application of Simulation: A Survey of Industrial Practice". *Interfaces* **13**, October: 47-52.

Clark, M.F. 1988. "WITNESS - Unlocking the Power of Visual Interactive Simulation". Paper to *International Conference on Visual Interactive Modelling,* University of Warwick, UK.

Clementson, A.T. 1973. *The New Extended CSL*. University of Birmingham, UK.

Clementson, A.T. 1981. "Fitting a Gallon into a Pint Pot". *Journal of the Operational Research Society* **32**: 319-326.

Collcutt, R. 1965. *The First 20 Years Operational Research,* BISRA, London.

Crookes, J.G. and Valentine B. 1982. "Simulation in Microcomputers". Journal of the Operational Research Society. **33**, 855-858.

Crookes, J.G . 1983. "Simulation on Microcomputers". *Paper to Operational Research Society Seminar,* Manchester, UK. February.

Cummings, N. 2008. "A little OR History, 1937-1994". *Inside O.R.* January: 16-17.

Dahl, O. and K. Nygaard. 1965. *SIMULA - A Language for Production and Description of Discrete Event Systems – Introduction and User's Manual.* Norwegian Computer Centre publication No.11, May.

Elder, M. 1995. Why SIMUL8? *Strathclyde Business School Research Paper 1995/13.*

Fiddy, E., J.G. Bright, and R.D. Hurrion. 1981. "SEE WHY - Interactive Simulation on the Screen". *Proceedings of the Institution of Mechanical Engineers,* ref C293/81. Institution of Mechanical Engineers: London, pp 167-172.

Fildes, R. and J.C. Ranyard. 1997, "Success and Survival of Operational Research Groups", *Journal of the Operational Research Society* **48**: 336-360.

Gilman, A. and R.M. Watramez. 1986. "A Tutorial on SEE WHY and WITNESS". In: *Proceedings of the 1986 Winter Simulation Conference,* edited by Wilson J.R., Henriksen J.O., and Roberts S.D, 178-183. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gordon, G. 1962. "A General Purpose Systems Simulator". *IBM Systems Journal* **1**: 18-32.

Head, A.E. 1962. *Handbook on the use of the Montecode Mark 2 interpretive program.* Report OR/CA/39/62, British Iron and Steel Research Association: London.

Hills, P.R. 1965. "SIMON: A Computer Simulation Language in Algol". In: Hollingdale SH ed. *Proceedings of the NATO Conference on Digital Simulation Operational Research, Hamburg*. English Universities Press: London, 105-115.

Hills, P.R. and T.G. Poole. 1969. "A Method of Simplifying the Production of Computer Simulation Models". *Paper presented at the Institute of Management Services Conference, Atlanta, Georgia.*

Hollocks, B.W. 1982. "FORSSIGHT is Better than Hindsight". *Paper to the North West Group of the OR Society*, Manchester. February.

Hollocks, B.W. 1992. "A Well-Kept Secret? Simulation in Manufacturing Industry Reviewed", *OR Insight* **5**: 12-17.

Hollocks, B.W. 2006a. "A Glimpse of Early OR - A Brief History of Cybor House". *OR Insight* 19: 18.

Hollocks, B.W. 2006b. "Forty Years of Discrete-Event Simulation: A Personal Reflection", *Journal of the Operational Research Society*, **57**, 12: 1383 -1399.

Hollocks, B.W. 2008. "Intelligence, Innovation and Integrity — KD Tocher and the Dawn of Simulation", *Journal of Simulation*, **2**, 128–137.

Hurrion, R.D. 1976. *The design use and required facilities of an interactive visual computer simulation language to explore production planning problems.* Ph.D. thesis, University of London.

Hurrion, R.D. 1981. "Visual Interactive Simulation Using a Micro-Computer". *Computers in Operational Research*. **8**: 267-273.

Insight International 1984. *Overview of OPTIK.* Promotional literature, Insight International Ltd, Woodstock, Oxford.

Kemeny, J.G. and T.E. Kurtz. 1964. *BASIC instruction manual*. Dartmouth College, Hanover, NH.

King, C.B. 1996. "Taylor II Manufacturing Simulation Software". In *Proceedings of the 1996 Winter Simulation Conference*, edited by J.M.Charnes, D.J.Morrice, D.T.Brunner, and J.J.Swain, 569-573. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Klein, A. 1988. "Factory Simulation Comes of Age". *Managing Automation,* August: 24-29.

Laski, J.G. 1965. "On Time Structure in Monte Carlo Simulation". *Operational Research Quarterly* A **16**: 329-340.

Law, A.M. and S.W. Haider. 1989. "Selecting Simulation Software for Manufacturing Applications: Practical Guidelines and Software Survey". *Industrial Engineering* May: 33-46.

Lines, A.H. 1981. "The Micro Computer in Management". *Journal of the Operational Research Society* **32**: 292-302.

Mathewson, S.C. 1974. "Simulation Program Generators". *Simulation* **23**: 181-9.

Mathewson, S.C. 1977. *A Programming Manual for SIMON Simulation in FORTRAN*. Imperial College, London.

Mathewson, S.C. 1984. "Simulation Program Generators: Code and Animation on a PC". *Paper to Operational Research Society Conference on 16-bit Micro-computers,* November.

Mellor, P. and K.D. Tocher. 1963. "A Steelworks Production Game". *Operational Research Quarterly* **14:** 131-135.

Mowery, D.C. 2003. "50 Years of Business Computing: LEO to Linux". *Journal of Strategic Information Systems* **12**: 295-308.

Naylor, T.H., J.L. Balintfy, D.S. Burdick, and K. Chu.1966. *Computer Simulation Techniques.* Wiley: New York.

Neate, R. and W.J. Dacey. 1958. "A Simulation of Melting Shop Operations by Means of a Computer".

In: Proceedings of BISRA Conference *Digital Computers in the Steel Industry*. Report OR/CA/4/58, British Iron and Steel Research Association: London.

Nolan, P.J., G.M. Lane, and J.M. Fegan. 1991. "ISI - An Environment for the Engineering Use of General Purpose Simulation Languages". *Simulation* **56**:.41-47.

Pidd, M. 1988. "Choosing Discrete Event Simulation Software". *OR Insight* **2**: 22-23.

Ranyard, J. 1981. "Introducing the Micro Computer Into the OR Department". *Journal of the Operational Research Society*, **32**: 277-296.

Robinson, S. 2005. "Discrete-Event Simulation: From the Pioneers to the Present, What Next?", *Journal of the Operational Research Society*. **56**, 6 (June), 619-629.

Sasieni, M,, A, Yaspan, and L. Friedman. *Operations Research: Methods and Problems*. John Wiley & Sons, New York. 1959.

Shanehchi, J. 1985. "EXPRESS: A Man-Machine Interface for Simulation". In: Heginbotham WB ed. *Proceedings of the First International Conference on Simulation in Manufacturing, Stratford-on-Avon*. IFS: Bedford, UK, pp 97-104.

Szabo, L.J. and D. Lyons. 1971. "Manual Simulation Technique". *Work Study and Management Services*, March: 196-198.

Tocher, K.D. 1952. "Report of a Meeting on Marshalling and Queuing". *Operational Research Quarterly* **3**: 12-13.

Tocher, K.D. 1960. *How to achieve optimum use of Appleby soakers.* Report 71/AF2/2, United Steel Companies Department of OR, Sheffield .

Tocher, K.D. and D.G. Owen. 1960. "The Automatic Programming of Simulations". In: Banbury J and Maitland J eds. *Proceedings of the Second Conference of the International Federation of O. R. Societies, Aix-En-Province.* English Universities Press: London, pp 50-68.

Tocher, K.D. 1963. *The Art of Simulation.* English Universities Press: London.

Tocher, K.D. and D.A. Hopkins. 1964. *Handbook of the GSP Mark II*. Report 118/ORD/10/TECH, United Steel Companies' Department of Operational Research: Sheffield, June.

Tocher, K.D. 1979. Keynote Address. In *Proceedings of the 1979 Winter Simulation Conference*, edited by H.J. Highland, R. Shannon, and M.G. Spiegel, 641-654. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Van Breedam, A., J. Raes, and K. Van der Velde.1990. "Segmenting the Simulation Market". *OR Insight* **3**: 9-12

Williams, J.W.J. 1962. "ESP: The Elliott Simulator Package", *Computer Journal* **5**: 328-331.

AUTHOR BIOGRAPHY

BRIAN HOLLOCKS is in the process of retiring as Professor of Management Science at Bournemouth University whom he joined in 1994 as the INTEL Chair in Computer Supported Co-operation. His lecturing and research included simulation, CSCW, knowledge management and AI. Prior to being an academic, he was in business and industry, primarily in OR and software development/marketing. He is a Mechanical Engineering graduate and holds a PhD in Management Science from Southampton University. His email address is: brian.hollocks@gmail.com.