INVENTING SYSTEMS ENGINEERING

John Aris

UK

Introduction

The year is 1949. Your Board has just approved an ambitious new project called LEO. You are responsible for creating the software (as we would now call it) for a series of substantial business applications for a computer. They include a sophisticated payroll, innovative stock control and production scheduling, and a remit to improve the business in whatever other ways you can make feasible. The hardware does not exist. Nobody in the world has ever specified or designed a business application for a computer. Nobody has ever written a business program. You have two years in which to work out the principles, design the jobs, get them programmed, make them work, ensure that they yield an economic return, and satisfy their users - all from scratch.

The man who was faced with this interesting challenge, and who met it triumphantly, was David Caminer, Manager of the Systems Research Office of J. Lyons & Co. Ltd. This paper is based on, and frequently quotes, conversations and correspondence with him. Direct quotations from him are indented throughout what follows. The author also gratefully acknowledges valuable suggestions from another early LEO pioneer, Professor Frank Land.

Fuller accounts of the LEO project will be found in two books on the subject: LEO - The Incredible Story of the World's First Business Computer[1], which contains a 150 page history of LEO by David Caminer, with contributions by twelve further authors who were personally involved; and the similarly titled LEO, the First Business Computer[2], in which Peter Bird tells the story with more emphasis on the engineering aspects.

A difficulty in writing about this early period is the terminology. At the time the terms for everything from requirements scoping to coding and operating were "systems and programming" for the activity and "programmer" for the practitioner. The activity «systems and programming» also included liasion with the end users (the clients), and helping to implement the system in the work place, though this activity drew heavily on the skills of the O&M Office.. In this paper I am using "systems engineering" and "systems engineer" in this same comprehensive sense; "systems analysis" for the phase of scoping, requirements analysis and specification; "systems design" for the phase of turning a requirements specification into a set of defined programs with detailed input layouts, file layouts, output layouts and technical provisos (such as reconciliations to be carried out); and "programming" for the phase of coding, testing and integration. The phases of course tended to overlap. I hope those who use these terms in rather different senses will indulge me. It should also be mentioned that throughout this paper I use the term "LEO" to refer to the project, the machine or the staff: the context will make clear which is intended.

---

1. LEO - The Incredible Story of the World's First Business Computer, by David Caminer, John Aris, Peter Hermon and Frank Land, McGraw-Hill, New York, 1998. The UK edition, published by McGraw-Hill, Maidenhead in 1996, was titled User-Driven Innovation.

2. LEO, the First Business Computer, by Peter J. Bird, Hasler Publishing, Wokingham, 1994.

Background[3]

It is remarkable that J. Lyons, a catering and food manufacturing company, should have initiated such a project, the first of its kind in the world. There was however good reason. The computer project was a natural, though ambitious, development of Lyons's pioneering approach to office work since the 1920s.

Lyons's main business activities were running restaurants and teashops, and making food products for sale both in the catering establishments and to other caterers and the public: principally bakery products, tea and ice cream. The business, by its nature, involved a very large number of low value transactions, and the administration for these was laborious, tedious and relatively expensive, to a degree which threatened the health and growth of the company. Lyons had always invested in modern and efficient manufacturing processes; in the 1920s they set their minds to a similar policy of investment in modern and efficient administration.

The mainspring of this was John Simmons, a Cambridge Wrangler whom Lyons engaged in 1923. Under his leadership during the 1920s and 1930s all Lyons's administrative processes were studied with great intellectual rigour, redesigned and redesigned again. The objective was always that they must be as effective and as economic as Simmons's impressive team, together with the best available technology, could make them. Lyons, always under the imperative of their multiplicity of low value transactions, became world leaders in this process engineering, then known as Organisation and Methods (O&M). Systems analysis and systems design (though of course not computer systems analysis and computer systems design) were integral parts of it. Pioneering applications of calculating and accounting machines and a world first use of microfilm were typical of the systems that were devised. Simmons declared, as a pre-LEO lesson from the past, "record keeping was of greatest value when it showed what ought to be done, rather than merely what had been done."[4]

It was therefore not amazing that it was members of Simmons's team who in 1947 were the first in the world to see that the "electronic brains" invented during the Second World War might have a role to play in business administration. Not only did they have this wild idea, they followed it up and established its feasibility, leading to the Lyons Board's remarkable decision in 1949.

The LEO project set up by that decision was to design and build a computer based on the Cambridge University EDSAC (which had triggered the Lyons decision by carrying out its first successful run that very day), and to mount on it a series of systems which would improve further, in effectiveness and economy, upon those already honed by twenty years of high-grade O&M work. They were to include payroll, stock control, sales invoicing, and whatever else could fruitfully be devised (the feasibility studies had even envisaged word processing, which in the event did not become economic for thirty years). This was not an experiment or a pilot project: it was expected to pay its way.

"There was no question of imposing applications. They had to be "sold" to the client departments."[5]

The key figures in the LEO project were Simmons, with Board level responsibility; Raymond Thompson, on whose ideas it was founded and who was in overall charge; John

---

3. The background is examined more closely in Hendry, J. The Teashops Computer Manufacturer (Business History, Vol 29, No 8, 1986) and in Land, F.F. The First Business Computer: a Case Study in User-Driven Innovation (Proceedings of the Kyiv Symposium, 1998).

4. Simmons, J.R.M. LEO and the Managers (Macdonald, London, 1962) p 25.

5. Note from Caminer to Aris, May 1998.

Pinkerton, who was to design and build the hardware; and David Caminer, who was to design and implement the applications. This paper concentrates on Caminer's area of responsibility.

The Challenge[6]

"The job I was given was to prepare production work for the computer system and to have the work ready to run as soon as the not yet fully designed equipment was itself ready to run it. Everything - coding standards, program construction and systems documentation - had to be created from scratch. The aim was always to be ready to use whatever facility Pinkerton was bringing to fruition."[7]

David Caminer had joined J. Lyons as a management trainee in Simmons's team aged 21 in 1936. He served in the infantry in the Second World War and then rejoined Lyons with the responsibility of assembling the management accounts for the Board - a task which (then as now) involved dealing with issues of incompatibility in terminology and data, of error trapping, of interpretation and of politics. He then became Manager of the Systems Research Office, which as its name implies was a forward-looking team, some twenty strong, considering new systems approaches and new technologies. He was thus the natural choice to take responsibility for LEO's Systems and Programming - though he was not released to do the job full time until May 1950.

In dealing with the challenge outlined at the beginning of this paper he had a number of important advantages. He had been trained in Lyons's elite O&M team. He had in depth personal experience of the company: its systems, its people and its culture. He was able to recruit to the project other individuals with a similar background, experience and knowledge (though not, in the early days, very many of them). His time in the Army had added further experience of the benefits of "good order and military discipline" when tackling demanding objectives. And he had the trust and support of Simmons and Thompson, and through them the trust and support from the company as a whole which Simmons's team had built up.

Perhaps, on the basis of that, the challenge sounds not so daunting after all? Just a matter of applying the well-understood Lyons approach to systems analysis and design, and coding the results? Hardly.

"Apart from the basic principles and objectives it emerged that there was little relationship between the analysis for a job using conventional equipment and that for an integrated application on the computer. What had started as draughts had become three dimensional chess."[8]

One obvious difficulty was that the systems to be implemented were already, in their non-computer form, highly developed. There was no fat on them - but economic savings were none the less required. The computer was notionally a wonderful tool, but copious data was going to have to be entered into it, and copious results were going to have to be printed. These operations would not only hold back the computer's electronic speed, but would introduce staff costs, particularly for data preparation, which were absent in the non-computer systems. Net savings were not going to be easy to achieve.

A less obvious difficulty was that a prime advantage of the computer was integration - tackling business processes end-to-end, without human intervention once the basic data was entered -but just how far should that principle be taken? where did the law of diminishing returns set in, and where did feasibility give out? There was no basis, either of experience or of theory, for answering those questions.

---

6. See Caminer's own (personally modest) account in LEO and its Applications: the Beginning of Business Computing (Computer Journal, Vol 40, No 10, 1997).

7. Note from Caminer to Aris, May 1998.

8. Note from Caminer to Aris, May 1998.

A third difficulty arose from the very absence of human intervention which was being targeted. Humans are intelligent and flexible. When something looks strange they notice it, and perhaps correct it or perhaps raise a query. Computers are not like that. The whole strategy for detecting and dealing with data errors was going to have to be rethought in minute detail and redesigned.

Fourthly, there was a group of difficulties arising from the computer itself. It had to be reconfigured halfway through the project because the original intention of using magnetic tape storage proved beyond the technology of the day. Its electronic speeds, though they seemed impressive at the time, were slow. Its internal storage was very limited. And its reliability was poor -a very serious issue when the daily operations of the business were to depend upon it: what could be done about this, and what contingency planning was required?

Fifthly, how were the new systems, once specified and programmed, to be validated? Computer time for testing was going to be a rare and precious commodity on a unique machine, and business users were not going to tolerate a trial and error experience when the systems went live.

Sixthly, programming was going to be an even more novel activity than systems design. The few computers that then existed were all in universities, research and defence establishments, and were used for mathematical and scientific tasks. Programming these tasks was a chore carried out by the mathematicians with a little help from the local computer team: it was not thought of as an important, or even an interesting, discipline in its own right (but mathematical skills were thought to be necessary for doing it). For most mathematical and scientific tasks the programs were relatively concise. Business tasks, on the other hand, were going to call for long, complex programs with complicated data and results, many alternative paths and much dealing with exceptions. Writing, testing and maintaining such programs was going to require discipline and professionalism. Professional mathematicians were not available to do it (and would have found the task deeply unsatisfying if they had been). And of course no software tools of any kind existed.

Some of these difficulties were obvious from the start, and solutions for them could be planned. Some gradually manifested themselves, and solutions had to be retrofitted to work already done. Some were apparent only to intuition, or could be solved only by intuition.

It might be asked whether experience of punched card systems took some of the sting out of the difficulties. Punched cards were something of a halfway house between non-computer and computer systems. Did they not throw light on integration? on coping with the absence of human intervention? on systems testing? on writing complex "programs"? The answer is no, because Lyons's experience of punched card systems consisted mainly of considering and rejecting them on cost-effectiveness grounds. In most cases they had found that the cost of data preparation and handling could not be compensated by savings from the limited integration such systems could offer. Whether if Lyons had had more experience of implementing punched cards this would have helped is an interesting question (the author's guess is yes, but not much).

Lines of Action

Faced by this situation, Caminer started four broad lines of action. Principles and standards for systems and programming work had to be laid down (and kept under review as experience developed); a team had to be assembled; research on how to write and test business programs was necessary; and the specification and design of the first applications had to be tackled.

Many of the early principles and standards, based on a combination of Lyons O&M experience and Caminer's very acute intuition of what was needed, lasted well:

- The scope and aims of each application must be searchingly and creatively established, defined and agreed with the client.

- Requirements specifications must be well written, well understood by the users, and agreed.
- Specifications should be kept as nearly frozen as possible during implementation.
- Systems should be as comprehensive as possible, dealing with exceptions as well as normal situations, but should not attempt "a bridge too far" (plenty of room for intuition here!).[9]
- System benefits must be explicit and quantified.
- An imaginative examination of possible pitfalls of all kinds in the system must be made, and evading action taken.
- Detailed flowcharts of paths through the system must be drawn and kept up to date.
- Data must be entered only once.
- All data must be rigorously checked for credibility by program, rejected if incompatible, but allowance made for subsequent reinsertion, and amendable if in error.
- Form design, for both computer input and computer output, is crucial.
- Reconciliations on the computer's, and the program's, internal workings should be calculated and displayed (this practice was eventually discarded in the early 1960s, by which time reliability had greatly improved).
- Programs must be carefully checked by a second programmer.
- Before trying them on the computer: the computer's time must be regarded as a means of verifying correctness, not finding errors (this principle is perhaps still valid, though cheap computer time has led to near universal neglect of it).
- No error is "the computer's fault": it is the fault of the people who should have corrected it, or allowed for the possibility of it happening.
- Spare space must be left in all programs for future modifications.
- Orderliness in thought and in documentation is essential.

Assembling the original team was not a huge task. It consisted of five people including Caminer, all (except one new management trainee) established Lyons staff.

"It is worth emphasising the extent to which Lyons did see the computer project as another venture in which the resources of the company would participate organically rather than as something deposited complete on the company from outer space."[10]

One of the team, Derek Hemy, had a particular aptitude for programming and program design. The first steps in LEO's programming research were undertaken by him in conjunction with the EDSAC team at Cambridge. A number of important ideas came from Cambridge, such as indirect addressing, and one of the Cambridge team sketched with Hemy the first outline of a payroll program, but as mentioned above mathematical programming was very different from business programming. Cambridge based their programming on the use of standard subroutines (eg square root, trigonometrical functions) which could be assembled into programs, whereas LEO, after one abortive experiment, made little use of this technique. LEO were interested in optimising store occupancy and run time by "clever coding", minimising the number of instructions written or to be obeyed, whereas this was never a priority for mathematicians. LEO emphasised the importance of breaking programs into discrete, comprehensible "stages" and annotating them so other programmers could more readily understand what they did - vital for both program checking and subsequent modification. Coding sheets, store layout sheets, file layout sheets and other documentation had to be designed and standardised.

The specification and flowcharting of the earliest applications was carried out by Caminer personally: a notable historic first. The use of flowcharts was itself innovative, having been

[9]. See Caminer, D.T., ...And How to Avoid Them (Computer Journal, Vol 1, No 1, 1958).
[10]. Note from Caminer to Aris, May 1998.

introduced as a system design aid by the Systems Research Office shortly before. Caminer had the help of Lyons O&M, who of course had not gone away and conducted most of the liaison between the LEO project and the users. Hemy also turned out to have a talent for estimating program sizes and timings which was invaluable in the systems design phase.

"What could go wrong was as important to identify as what was needed to go right. Errors and stoppages had to be guarded against, not accepted as inevitable, for all that the art was in its infancy."[11]

The principle was to establish the outlines of a system first and then to amplify them through successive levels of detail. The typical sequence was that Caminer, with his thorough knowledge of Lyons processes, drafted the scope and aims of each system and agreed them with user management. He then went on to detail inputs, outputs, calculations to be performed and any other key issues, and to draft a (necessarily long and elaborate) requirements specification, including outline flowcharts, which with the aid of O&M was agreed and signed off by the users. Inputs and outputs were now frozen, and with the help of Hemy the computer design was worked out. This would evolve during implementation as storage and run time considerations became clearer.

The First Applications

It had always been assumed that LEO's first application would be the Lyons payroll. However as the project progressed three further major jobs were developed almost in parallel with it. They were called Teashops Distribution, Reserve Stores and Tea Blending. All these broke new systems ground. Teashops Distribution was a time critical daily stock replenishment system for some 200 retail outlets. Reserve Stores was a stock control and production scheduling system. Wholesale Bakery Sales and Accounting, was a highly integrated set of procedures involving sales despatching,and invoicing, sales accounting, sales statistics, directions to packers on carton sizes, packers bonuses and management reports. The output from the application went to the thousands of small and large retail stores selling food products throughout the land. It was probably the first computer job in which the public received and used computer output as part of their everyday work. Tea Blending not only tracked stocks of about 300 types of unblended tea, available or in transit from growers, but provided for the first time full, detailed cost and availability data for the complex problem of mixing these into branded blends: effectively a decision support system.[12]

The first system to be specified in detail was the payroll. In the feasibility phase pre-1949 the outlines of a computer payroll system - inputs, brought forward and carried forward files, outputs - had been laid down (there is a perfectly valid half page description in the 1947 report which inspired the LEO project), but Caminer now had to go into all the massive detail which would put flesh on those bones. What should be in those files, and how much file space would it require? exactly what input data, and exactly what outputs, were needed? what must the input and output forms look like? precisely what credibility checks must be carried out on the data? exactly what calculations must be done (for example, what tax calculations on advances of pay to those going on holiday)? what exception routines were needed? what aspects of payroll might be "a bridge too far" (in the event, only team bonus payments)? what possible pitfalls might there be, in the logic of the system, in its implementation, in the regular operation of the computer, in the surrounding manual procedures? and what should be done about them? what reconciliations were needed? what contingency plans? what management information might be a valuable by-product?

---

[11]. Note from Caminer to Aris, May 1998.

[12]. These systems are described in Land, F.F. Systems Analysis for Business Applications (Resurrection, Summer 1996).

and so on. Specifications for non-computer payrolls had of course been written before, but many points had to be covered in this one that no previous one had addressed.

The payroll was the first such specification, but Teashops Distribution was not far behind. Unlike payroll, this was not an adaptation of a pre-computer system, and had not been an explicit part of the LEO remit: it was an entirely fresh look at an old but recalcitrant problem. The problem was that each of the 200 teashops had to reorder hundreds of bakery and kitchen products every afternoon for the following day, to ensure freshness and avoid waste (these were the days of rationing). The products then had to be made, packed and delivered overnight. The raising of the orders was a big and tedious task every day for the teashop manageresses. Once their order forms were delivered (physically) at Lyons's Hammersmith works there was formidable paperwork to be done to initiate production and despatch, and the time available was short. It was certainly too short to allow a big data preparation operation followed by a series of computer runs before production could begin. At the same time the computer was potentially an ideal way of optimising production runs, assembly of individual teashops' orders, van loading, and the necessary documentation. The new system was Caminer's invention. He discovered, in true O&M fashion by studying piles of previous transactions, that orders for any one teashop fell into a pattern dependent on the day of the week. Armed with this knowledge, he devised a system whereby the computer produced for each manageress each day a suggested reorder. The manageress now had only to indicate any changes she wanted to make, and the data preparation load was vastly reduced. The manageresses dictated any changes over the telephone direct to keypunch operators (such use of the telephone was itself innovative in the early 1950s!), the resulting punched cards went into the computer in batches, and the computer could do its part in good time for production. Not quite a real time job, perhaps, but very much a time critical one: probably the first of its kind in the world by at least five years. Most computer applications throughout the 1950s processed data after the business event, rather than on the critical path before it.

"It all seemed so obvious that what Simmons described as the "incredible speed" of the computer would have to be matched by a medium that was equally state of the art. The fact that with planning and programming skill a balanced system could be put together with paper tapes as the main current data medium and with punched cards as the main data storage medium had not been explored."[13]

The LEO team had initially envisaged magnetic tape as the main device for carrying data, and magnetic tape drives were installed in the early days. However, the technology chosen proved unworkable at that time, and punched card equipment was installed instead. The set-back with magnetic tape delayed the major applications significantly, so they were not in fact the first jobs to run on LEO. Three types of work were carried out appreciably earlier, and were important ingredients in the evolution of systems engineering. They were the test programs for the computer, a series of mathematical jobs undertaken on an opportunist basis for customers outside Lyons, and the Bakery Valuations job which had the distinction of being the first regularly run (though not major) business computer application.

It became clear early in the project that test programs were a non-trivial requirement. This computer, uniquely for its time, was going to have to give dependable service to a business. Teashops Distribution was particularly time critical, but all the jobs had a degree of time criticality. Payroll had to be ready (and correct) for payday, tea stock reports had to be available to production planners when needed, and of course delays anywhere were likely to be very damaging to the management credibility which LEO crucially needed to sustain. However 1950 valve based technology was inherently fault prone. Specifying and programming the test programs became a challenging application in its own right. The detailed requirements were worked out and

---

[13]. Note from Caminer to Aris, May 1998.

documented (and continually updated as experience accumulated), flowcharts were drawn, programs were coded and independently checked. Each logical function and each circuit had to be exercised. Though this application bore little resemblance to a business system it was sophisticated, and its calls on systems engineering skills were an important stage in their evolution.

"It scarcely mattered whether a scientific job was executed today or tomorrow or the day after. It was generally being accomplished altogether faster and with less human effort than had hitherto been conceivable."[14]

Mathematical jobs came in when it became known in the market that the computer under development could offer some capacity, particularly for work which was not time critical and had no need for bulk input or output. Typical jobs were weather forecasting, ballistic tables for the army and flutter calculations for aircraft design. These jobs were again subjected to, and influenced, the evolving systems and programming disciplines, and furthermore were operated in a revenue-earning context. This first experience of live operation cast new light on the issue of hardware unreliability and how to deal with it. Restart procedures were introduced, which allowed long runs to be repeated in part rather than as a whole if they had encountered hardware trouble. It sharpened thinking and sophistication for the design of in-program reconciliations. It also illuminated issues of accuracy in rounding off large numbers.

The Bakery Valuations job was a modest one, again not demanding bulk input and output and not especially time critical. It was scheduled at Caminer's instigation when it became clear that the major applications would have to be postponed while new input/ output mechanisms were designed and developed to take the place of magnetic tape. There was some resistence from above to introducing such a trivial job: not the kind of major advance that Lyons were looking for from their LEO investment. None the less it was useful, it provided economic savings, it would yield for the first time experience of regular running, and it would be a satisfying and tangible achievement for the systems engineering team. Caminer, again, specified and flowcharted the system, which valued the weekly output of each bakery, bakery despatches to each sales outlet, and stocks awaiting despatch. The results went to top management. Reconciliations and restart points were built into the programs. It went live in November 1951 and ran every week thereafter. It saved money and produced its reports faster than the previous non-computer system. Perhaps more significantly, it provided experience of using data from the outside world, with all the fallibilities which had previously been circumvented by intelligent human interpretation on the job. It also turned out to be a valuable extra test program for the hardware - and even contained an alternative piece of code to be activated if one particularly refractory circuit was malfunctioning!

Implementation Tasks

Programming the early applications, particularly the major ones, faced three particular challenges. The first was retaining an overall intellectual grasp, while coding in machine code (though not, mercifully, binary), of the sprawling complexity of the programs. Flowcharts, breaking the program into short, logically coherent stages, and annotation were vital to this. The second was fitting each program and its data into the store (there was of course no backing store). This required minimisation of the number of instructions in the program and tight packing of data. Data had to be packed more tightly still on the brought forward/carried forward files, held on the slow medium of punched cards. The cards were ingeniously used to their utmost by holding binary numbers horizontally rather than decimal numbers vertically: this was one of the many clever expedients devised as the project proceeded to cope with the limitations of the hardware. Cards used for the entry of data were used in the conventional vertical mode. Another was forming instructions by program for later execution ("if X, plant an instruction to go to Y ten steps ahead"). The third

[14]. Note from Caminer to Aris, May 1998.

challenge was to minimise run time. Major file processing programs, and major mathematical tasks, took hours not minutes to execute. As computer time was scarce and valuable, and as the mean time between faults was short, reducing these run times was a priority. One instruction saved in a main loop could reduce overall running time by many minutes. Unfortunately it was often only possible to save running time at the cost of increasing store occupancy, and vice versa. Satisfactory compromises could sometimes be achieved only by redesigning the program structure of the system.

These challenges will be unfamiliar to the great majority of today's programmers. They made programming much more interesting but were expensive in skilled people's time - which however was cheap in relation to computer time. An interesting reflection of the skill employed is that the run times for the payroll, estimated on the assumption that the file holding medium would be magnetic tape, were bettered in practice despite having to use much slower punched cards.

Independent checking eliminated many program errors, but inevitably not all. Debugging on line was discouraged because of the scarcity of computer time, but none the less some expertise in it was developed, usually in the small hours. It involved peering at binary patterns on a cathode ray tube, with some additional help from being able to hear patterns of machine activity on a loudspeaker. The alternative was to study store dumps, also in binary, off line. Test data came in two varieties. First the programmer devised his or her own, trying to cover all paths through the program and all permissible and impermissible data variants. When he or she was satisfied, real data, more voluminous and typical but less logically stringent, was acquired from the prospective users for a further round of tests. Only when all glitches revealed by either set of test data were fully explained and corrected could pilot runs with the users begin.[15]

LEO People

"The rapidity and the success of the achievement owed much to the preparatory thinking that had anticipated the decision to proceed, but a great deal, too, was due to the closeness of the working relationship between myself and John Pinkerton."[16]

A feature of the whole project was how closely the "systems and programming" and engineering teams worked together. There was little difficulty in doing so, as both teams were very small, largely made up of people with Lyons experience, and located together. The benefits were significant. The "programmers" were able not only to influence the hardware's logic design (for example by calling for hardware instructions for conversion between binary and decimal), but also to understand issues such as circuit reliability which directly affected their own work. Similarly the combination in the same individuals of the systems analysis, systems design and programming roles led to an optimisation - which was highly necessary given the limitations of the technology. The loss of this close interdisciplinary cooperation, though no doubt inevitable as the computer industry grew, has been sad.

It was clear that for regular, time critical business applications to succeed, hardware and software were not enough: professional data preparation and professional computer operating would also be required. For data entry two media were used: punched paper tape, where variable length fields were needed, and conventional (decimal) punched cards, where mechanical sorting was a requirement. For punched cards the discipline of keypunching, checked by rekeying and comparison, was well established and the appropriate equipment was available. This was not the case for paper tape. LEO designed a paper tape "comparator" specifically to carry out the checking. A team of punch operators was assembled: this was the world's first computer data preparation

---

[15]. Fascinating material on the early days of programming, supplied by Derek Hemy, will be found in Bird's book (see note 2), pp. 52-62.

[16]. Note from Caminer to Aris, May 1998.

section. Operating again presented new challenges. Nothing in the nature of operating system software existed. Much manual intervention between programs, often including card sorting and printer plugboard changes, was required. Decisions had to be taken when unexpected stoppages occurred, as they often did. Again a specialised team was formed.

As the reader will have realised, the work on the major and minor applications, and the ever-increasing understanding of systems engineering, was intellectually fascinating and absorbing. It was also extremely demanding. The team frequently worked all night. Caminer was a hard taskmaster and his wrath was terrible - though it was reserved for occasions when it was deserved (cutting corners, ignoring disciplines, making the same mistake twice), and he worked himself as hard as the others.

"I believe that I was regarded as unreasonable in those days. There seemed no other way. There was so much to do and so little time to do it in, and resources were so limited. To keep ahead of expectations in Systems Engineering and to produce a stream of dependable, economically viable integrated jobs meant years of grinding, dedicated work by the application team. I myself was completely dedicated to the work in hand and placed it before family commitments and other interests. I expected others to behave similarly on the particular jobs with which they were concerned, and to be ready to help out in a comradely manner when one of their colleagues needed a hand. When they had been engaged they had been warned that the work would be both mentally and physically exacting. That had frightened no one whom we wanted away. But looking back it must have been a sore trial. My own wife was totally resilient in dealing with my odd hours and absences and our young children saw little of me. Happily the other young wives were tough too, though it must have been less easy for them to understand the voracious demands of what we were about."[17]

The team now began to be expanded, with individuals drawn from within Lyons but soon also with new graduates, and questions of how to select and train them had to be considered. This gave rise to an aptitude test (essentially a simple exercise in programming) and a training course, both designed by Caminer, which ran regularly (with occasional adaptations as the technology developed) for some ten years and were much praised by LEO's customers.

"Month-long training courses were established which converted newcomers in a short time into active contributors. The existing staff were the instructors and the evaluators. Students quickly became teachers."[18]

The Challenge Accomplished

The new input/output system came on line in 1953, and the first pilot runs of the payroll took place that June. Lyons were understandably cautious about committing anything as sensitive as the pay of their employees to a system as innovative and vulnerable as this, so a long series of pilot, then parallel runs took place. Parallel running, as its name implies, involved comparing the computer's results in detail with those produced by the non-computer system and explaining any differences (by no means always errors by the computer system!). Though costly and tedious, this process was essential not only to getting the computer system right but to building management's confidence in it. Parallel running was an accepted Lyons practice for non-computer systems, but once again it was the first time that system proving on such a scale had been done. Eventually the first live run took place in February 1954, and 10,000 employees were on the system by the summer of that year. Lyons would not go beyond that number while there was only one computer, and contingency plans for making emergency payments remained in being for some years. However,

[17]. Note from Caminer to Aris, May 1998.
[18]. Note from Caminer to Aris, May 1998.

they were never needed. The system ran every week, and though of course there were both hardware and software problems from time to time it never failed to deliver.

Teashops Distribution was not far behind. It went live in October 1954, and ran regularly and dependably every day thereafter. Its inventiveness, and in particular the work it saved the manageresses every day, were quickly recognised and acknowledged. After only a few days the daily report from the Wembley teashop said "the head staff at this shop would like to give thanks for LEO. This is a wonderful time-saver, work-saver, and we are grateful for it." As the manageresses were a notoriously crusty group this was a most satisfying achievement. Reserve Stores and Tea Blending also successfully went live during this period. Reserve Stores was a short-lived job because its rationale went away with the abolition of rationing, but Tea Blending continued to run (on successive generations of hardware) for at least 25 years and may still be in operation in recognisable form today -the world's oldest business application?

The LEO team, therefore, had gone from the completely green field of 1949 - no hardware, no experience anywhere of computer systems analysis, computer systems design, or business programming - to five major live applications and a large number of minor ones in a few months over five years. But for the magnetic tape problems that time could have been considerably shorter. The major applications were sophisticated even by 1998 standards. They were economically successful and satisfied their users. The number of people who had achieved all this was extraordinarily small: up to the go-live of Bakery Valuations fewer than 20 people in total (management, design and maintenance engineers, "programmers", secretaries, data preparation staff) had taken part. By 1954 the systems engineering team had risen to about ten, none of them of course with any previous experience.

Spreading Beyond Lyons

This is an impressive story, but of course it is only the very beginning of the story of systems engineering. The leading role in that story continued to be LEO's for at least another five years, as a number of striking new developments now took place. Further major applications for Lyons were initiated. Outside customers, impressed by what Lyons had achieved, began to commission systems from the LEO team. Lyons not only approved the building of a second machine for themselves, but agreed that LEO could make machines for customers. The teashops company was in the computer business. And the systems engineering principles and practices developed during the formative period turned out to be generally applicable.

The first major outside customer was the Ford Motor Company. They knew and respected Simmons through the Office Management Association (now the Institute of Administrative Management), of which he was a leading light. They came to see the Lyons payroll - and commissioned a payroll of their own to run on LEO's machine. Obviously they could not use the Lyons software as it stood, but the principles were similar and quite a lot of the program code could actually be copied. Caminer produced a draft specification in May 1955, the order was placed in August, parallel running started in November and the first live run was in December. For a major custom-built system this timescale still seems incredible.

Many other customers followed, initially for jobs to be run on LEO's machines (the second went live in May 1957), and then for machines of their own. The first customers had strong O&M teams of their own, and an approach similar to that at Lyons was possible, with the LEO systems engineers collaborating with the customers' O&M. Caminer personally specified the Stewarts and Lloyds (a steel firm) payroll and others. Indeed members of the LEO team were still writing specifications and designing systems for customers - often in the form of elaborate pre-sales proposals - in the 1960s. However, a notable difference gradually became plain between customer and Lyons work: the reduced room for manoeuvre in systems analysis and design. In Lyons there was a general understanding that the efficiency and effectiveness of any business process could and should be improved, and that computerisation was an excellent opportunity for doing so. For many customers radical change in working practices was not an option: top management were in most cases not sufficiently committed to push such radical change through - even if they recognised it as desirable. They were therefore looking for something which saved money but did not aim at uncomfortable optimisation. In many cases they had a background in punched card systems, and saw a computer as an enhancement of unit record equipment rather than as a major, liberating innovation. LEO were able to satisfy such customers - but were always happiest dealing with the minority of adventurous customers who saw the computer's true potential. There were indeed customers who expanded LEO's thinking. One early example is that Lyons had resolved that alphabetic (in addition to numeric) printed outputs were an expensive luxury, and that all results should be coded as numbers. The first customer to buy a LEO machine, the tobacco manufacturer W.D.& H.O.Wills, saw the extra flexibility and user friendliness of alphabetic printing and insisted upon it.

In the 1950s and early 1960s adventurous customers for business systems may have been commoner in the UK than in the US. There was an important difference in the economics of computing between the two countries. In the US labour costs were high, so cautious computer systems, easy to implement but falling far short of exploiting computers' true potential, could readily be justified by quite minor staff savings. As a result, most early business systems there did not depart far from the pattern established using punched cards (much to the benefit of IBM). In the UK, by contrast, labour was cheaper so unadventurous systems were harder to justify.

The strategy by which LEO specified and designed adventurous customers' systems - it could be regarded as an early manifestation of outsourcing - was successful but could not be sustained indefinitely. Customers were encouraged to hire their own professional staff, often from LEO, and gradually took over these tasks for themselves. But by the late 1950s the majority of the world's sophisticated business computer applications were still LEO designed, or designed by systems engineers trained by LEO, in accordance with the principles laid down by Caminer and his staff - though the competition was by then beginning to catch up.

Commentary

The list of LEO world firsts in systems engineering is impressive, but perhaps a more important question is which of LEO's achievements lasted to become part of world best practice. It is also interesting to ask what, with hindsight, LEO got wrong.

The list of firsts goes like this:

- Lyons conceived the idea of using computers for business administration before anybody else.
- Caminer specified the first business applications of computers both in and outside Lyons and, with his team, designed the systems.
- these early specifications and designs pioneered flowcharts, file design, form design for input and output documents, inprogram reconciliations and restart procedures.
- more subtly, they developed the concepts of systems integration (entering data once and squeezing the most out of it) and of handling error-prone data without human intervention.
- they involved business process re-engineering, and included time critical and decision support systems as well as systems initiating operational action rather than reacting to it.
- Hemy established the disciplines for writing the long, complex programs characteristic of business systems, and with his team wrote the first such programs.

- LEO created the disciplines of checking programs off line, debugging them, and carrying out large scale pilot and parallel running.
- LEO set up the first professional computer data preparation unit, and the first professional computer operators.

What were the lasting achievements? By the time the author of this paper joined LEO (in 1958, some five years after parallel running of the Lyons payroll began, and a few months after the delivery of the first LEO machines to outside customers), there was a highly professional systems engineering department, including customer as well as LEO staff, constantly applying all the lessons of the early applications. Further experience was of course continually adding to good practice, but the principles were remarkably unchanged and survived the advent of magnetic tape, operating systems and other upheavals. Most of them remain sound today, though some are now concealed in software (eg operating procedures) and some have been overtaken by technological advance (eg the need for in-program reconciliations). Some are neglected though they should not be (eg flowcharting, off line program checking).[19]

LEO, and to a lesser extent its successors English Electric and ICL, continued to teach those principles and their further enhancements for many years. However other computer companies by the early 1960s were teaching their own versions, and the cohesion and discipline of LEO's approach gradually disappeared. Much of that approach was obvious good practice and was reinvented (and is still being reinvented) elsewhere. LEO's claim is not that its thinking was unique, but that it did it first and it did it whole-heartedly.

What did LEO get wrong in the field of systems engineering?[20] In the early days it went up some blind alleys (for example a very complex set of flowcharting symbols, or a program structure based on closed subroutines like those used for mathematical algorithms), but the practice that had evolved by 1954 was satisfactory. But perhaps, more importantly, LEO lacked the resources or time to try multiple approaches, experiment with these,and then back the winner. It was very dependent on high-quality individuals working extremely hard, which was not sustainable long term. LEO perhaps missed an opportunity to reuse systems and coding (moving towards software packages) - but the hardware constraints of space and time made anything other than intelligent Chinese copying infeasible until the late 1950s. LEO indeed got so good at Chinese copying that it missed the importance of forward compatibility, which IBM exploited so well in the 1960s. It was inward-looking: in an ideal world it would perhaps have deliberately spread its hard-won good practice to universities, business schools, software houses, consultants, other computer manufacturers.... but the time was not ripe, effort was scarce, and the task did not appeal, not least for commercial reasons. Some dissemination did take place through the export of LEO people, not only to customers but widely both in the UK and abroad.

Systems engineering was invented in many places in many ways, it must be admitted. It emerged from punched card, and other office machine, techniques in the manufacturers of those devices and in their customers, from engineering practice in engineering companies, from accounting practice in financial departments, from trial and error everywhere. This paper has set out to describe how David Caminer and his team did it very early, very successfully, through a combination of prior experience, intuition, and conscientious and imaginative learning on the job, and set a standard of best practice which was unique for its time.

---

[19]. Some comparisons between LEO and modern practice will be found in Aris, J.B.B. Systems Design - Then and Now (Resurrection, Summer 1996).

[20]. The wider question of why LEO did not sustain its world lead in business computing to become a dominant supplier is intriguing but beyond the scope of this paper. It is the subject of a so far unpublished note from Caminer, dated 19.7.98, to the author.