[1]Computer RESURRECTION

The Bulletin of the Computer Conservation Society

ISSN 0958-7403

Number 43 Summer 2008

Beacon 1963-7: A System Design Ahead of its Time?

Michael Knight

Michael Knight recalls Beacon - the pioneering reservation systems developed by British European Airways (BEA) and UNIVAC. Despite achieving so much with so little, Beacon did not survive the BOAC (British Overseas Airways Corporation) merger, but its ghostly legacy lingered on for a while in unexpected places.

Beacon was British European Airways' computer online network, initially developed in 1963-7, to provide a full-scale passenger reservations service. Subsequently, the hardware was upgraded, and further applications added on an integrated basis. Later, following the merger with BOAC to form British Airways, these services were progressively taken over by Beacon's old rival system Boadicea. The memory of Beacon thus began a long evaporation.

The purpose of this memoir, 40 years later, is to record something of the achievement of the Beacon project team on what was, for a time, Europe's largest and most ambitious business transaction-processing (TP) system. TP is the norm now, taken for granted in systems of any size, but back then was rare and leading-edge, not even called TP, but 'real-time'.

The first part outlines the task, and the resources deployed to implement it: hardware, software, time and people. The second part sets out to justify this paper's bold title, by examples rather than by a systematic description of the design. Some of these examples were subsequently reflected in standard operating system facilities for TP applications; some were not. In the latter case, this has been due to the awesomely rapid developments in computing hardware capabilities, which have allowed the evolution of increasingly function-rich and hardware-profligate operating systems and other software tools. In the early 1960s, before integrated circuits and the ubiquitous byte, when ferrite core memories were hand-knitted, programmers

[1] This article is based on a talk given by the author to the Computer Conservation Society (CCS) in London in February 2008. The talk was attended by several who had worked on this pioneering system. The CCS holds meetings in London and Manchester. The Society's 'Computer - Resurrection' bulletins can be browsed on their website, www.computerconservationsociety.org. This article is reproduced on the *VIP CLUB* web site with permission from the author and the Computer Conservation Society.

were far from cheap. But hardware was unimaginably expensive by today's standards, and limited in its capability, and there were definite limits to what could be had in one box, at any price.

The Task

In the mid-1960s, BEA carried over 7,000,000 passengers a year, and was 5th largest 'carrier' in the world. [BOAC carried over 1,000,000 a year and was ranked 35th largest]. By today's airline standards, these numbers are very modest, but they presented very demanding data processing challenges. The real-time solutions to these challenges were the ancestors of today's all-pervading call-centres[2].

Airline reservation systems involve up-to-the-minute updating of numerical records of flight occupancy - or, more accurately, of each flight/class/segment/date, in BEA's case for 11 months ahead. This was known as 'inventory'. Related to that, alphanumeric passenger record details need to be formed and maintained, a record being for one or more named passengers travelling together on an itinerary of one or more flights. This was known as a PNR - passenger name record. Linking PNRs to relevant flight records were Passenger Name Indices (PNIs).

Bookings, amendments, etc. were typically made by telephone, by airline or travel agents, or by travelers. An added complication was that airlines made bookings with each other, on behalf of travelers, and needed to be kept aware of the booking status of many of each others' flights, as well as traveler details. The medium for these 'interline' bookings and status updates was a jointly-owned international airline message network, SITA, carrying mainly manually-originated tele-printer messages on 75bps asynchronous circuits, the messages conforming [or more often not] to a common protocol called AIRIMP. In the '60s, about 60% of BEA's bookings were interline, i.e. through other airlines.

The reservations task did not much lend itself to the prevalent batch processing of sequential files, and was very labour[3]-intensive. Passenger records, for example, were maintained manually in card files of hand-written cards. To complicate matters further, flight schedule changes needed to be managed and, ideally, seat availabilities adjusted, through the months prior to a flight in the light of [not very reliable] statistical booking patterns.

To replicate all this work, with a computer system, gave rise to further requirements. It would need to be 'user-friendly' [a later buzzword], not least for hundreds of telephone sales clerks. It would need high performance - responses to transactions within 2 seconds - much faster than that was considered to be user-disconcerting. It would need to be extremely reliable and operational 24x7, being crucial to the airline's core business. In fact, the successful UNIVAC proposal and contract specified, by a carefully constructed formula and with attaching financial penalties, better than 99.97% system availability, around the clock.

When the project began late in 1963, there were a few precedents. American Airlines, with SABRE, and Pan American, with PANAMAC, were implementing similar systems with IBM.

[2] The American spelling of 'centre' is 'center'.
[3] Note, many words in England have an 'our' ending versus the American 'or' thus labour vice labor.

More relevantly, Eastern Airlines, Capitol and Northwest Airlines had completed numeric or 'inventory' implementations with UNIVAC, but not full PNR systems. In 1965, having lost at BEA, International Business Machines (IBM) won a system for BOAC, based on the delusion of a ready-made reservations application package and a serious underestimate of hardware requirements.

The risk and investment cost in these early systems were huge, hardware costs alone amounting to several £millions [in old money]. What were the business motives?

There were several, with varying degrees of plausibility, as the 'me too' fashion for such systems spread among the carriers. The original, most measurable motive among the major airlines was cost-reduction, meaning significantly fewer staff in the labour-intensive manual reservations function - which also tended to be the least unionized. This bottom-line benefit was easily measured and predicted.

Better load-factors, i.e. more bums on seats per flight, was another benefit likely from greatly improved inventory control, but less readily quantifiable, because of other contributory factors. So, too, were enhanced customer service and the advertised boast of leading-edge computer technology: expensive, but cheaper than new aircraft.

Hardware

The heart of Beacon's hardware complement comprised two UNIVAC 490 mainframe computers with fixed head and moving head drums amounting to 700 million characters.

The UNIVAC 490 had 32k words of 30-bit core memory, addressable as half, third or fifth words, the latter representing a six-bit character when required. The single-address instruction set was relatively limited, but powerful, including several 'replace' commands and the ability to test an arithmetic result in various ways and 'skip' or not the following instruction. Instructions had 15-bit addresses, modifiable by any of seven index [B] registers, while A and Q were arithmetic registers. Add time was 4.8 µsecs, and average execution time was six µsecs. There was no memory protection - seen as an advantage by performance-conscious 490 fans, provided programs behaved properly.

Normally, most of the peripheral equipment was attached to the currently on-line 490 computer. All subsystems were individually switchable between the on and off-line computers by means of manually operated switch buttons.

Over 400 'agent set terminals', of which about 200, in 1967, were installed in nine remote locations, were connected in groups via buffer processors, small minicomputers used as early cluster controllers. The agent sets, manufactured by Sperry Gyroscope, comprised a tele-printer, keypad and upright keypad/display, onto which an appropriate route schedule slide was inserted; the slide was hole-coded to transmit the route details. The tele-printer component was principally used in the second, passenger records phase of the project. In subsequent reservations systems, and in later Beacon applications, these relatively slow and cumbersome terminals were replaced with more flexible VDUs.

The hardware occupied an air-conditioned computer room of some 6,000 square feet on the third floor of the West London Terminal on Cromwell Road, West London. Most of the local agent sets were located on the fourth floor of the building, in an area of some 18,000 square feet, supported by automatic telephone call distributors.

A round-the-clock team of 15 UNIVAC hardware maintenance engineers watched over the installation, to minimise[4] breakdown times.

Software

The 490 came with REX, Real-time Executive; totally inadequate for Beacon's requirements. The system was therefore built on a home-made Executive, CONTORTS. Other significant project-developed tools included an interpretive on-line transaction trace, a general utilities system (GUS) and an online training mode for terminal operators.

Time

The hardware costs alone were huge; around £3M in 1963 - upwards of £40M in today's money {sic 208}. It was, therefore, important that the benefits were real and measurable, and that the return on investment began as soon as possible. This meant that the initial application, passenger reservations, had a *phased development*. First, the inventory, or numeric control of flights bookings, was developed and cut over to live operation, in April 1965, delivering the benefits of improved load factors and staffing efficiencies. Then the 'PNR', or passenger records processing, together with handling of interline traffic through the SITA network, was developed, and cut over in stages. A third phase extended the real-time service to numerous remote offices in the UK and mainland Europe, and eventually on-line ticketing was added. Subsequently, other related applications were added, on an integrated basis.

This may seem obvious, commonplace in Information Technology (IT) projects, even. Back then, it was not. Once the first phase was implemented, it was committed to the core of the airline's business - 24x7. Adding major enhancements securely to such an operational environment is not easy, particularly when, as with PNR, it involves the rolling transfer of millions of existing paper passenger records, entered by keyboarding, while those records continued to be subject to additions, cancellations, or amendments. Then add the requirement to train several hundreds of user staff in the additional features of the system, and to incentivise their contribution of huge amounts of overtime for the keyboarding task

The cutover of the second, PNR phase, occurred in 1966, and was a substantial project in itself, more organisational than technical. It was planned for Easter. Bank holidays were particularly useful, being long, and light in on-line reservations activity; plenty of time for expensive sales clerks' overtime, and sleepless nights for key development project staff.

It failed. The phase 1 implementation was sadly and successfully reinstated for the Tuesday morning. For various reasons, it failed again, every Bank Holiday until the end of the summer. Even then, subtle but stubborn bugs persisted including one which struck every morning at peak

[4] In Britain it is not unusual to spell some words with 's' versus 'z' used in America, these are not spell check errors!

time in the heart of COHORTS which added a ‘one’ randomly to the contents of a word anywhere in core memory.

People

The development project team, of around 25-30 people, was roughly evenly divided between UNIVAC and BEA staff. It was divided into teams responsible for specific parts of the application and system software, from three to six in each. Team leaders came from either company, appointed on the basis of perceived contribution, in terms of skill and effort. Until the reservations development was successfully completed and signed off, Univac retained system responsibility and project management, along with the challenging contractual obligations on performance and availability. The three-phase reservations development took about 80 man-years. This is somewhat misleading, since hours worked would become, for some, quite brutal, or heroic, [according to taste], and not respectful of public holidays, weekends or private life.

UNIVAC’s project management was vigorous, some might even say brutal. Mindful of the contractual obligations and of the finite project headcount, it was not tolerant of non-commitment or technical inadequacy, and this sometimes extended to BEA as well as UNIVAC team members. On the other hand, contribution was remembered and proportionally rewarded, sometimes long afterwards, by promotion and/or recruitment [in the case of several BEA staff], and/or recommendation elsewhere.

The philosophy, later articulated [within ICL!], was that the project goal [on- time, to specification and budget] was the only thing which mattered to the project team. Nothing, from fire on the site to strike action to bureaucratic obstruction in either organisation would be allowed to get in its way. If either host organisation found the project’s activity disruptive, well, they should not have instigated it; substantial projects are, of their nature, disruptive.

Systems analysts in BEA originally determined the requirements of the reservations system. During the implementation project, there were no specific analysts on the team; everyone was a programmer. [This changed subsequently.] The Univac management, extremely conscious of the trade-off between nice-to-have features and precious computing hardware resource, and of a limited team headcount, did not feel a need for middlemen, brokering the fine-tuning of requirements, between system end users and programmers. Indeed, one manager somewhat insensitively defined ‘analyst’, before an audience including several of them, as the opposite of ‘catalyst’: an agent which, while taking no active part in the process, nevertheless impedes its progress.

Design Features

Prepared by the forgoing background information, it remains to attempt to justify the presumptuous title of this memoir; firstly, with some basic design principles; secondly, with a selection of design features - and remembering that this was the mid-1960s.

Design for Failure:

The site's hardware engineering manager once asserted that hardware, of its nature, was bound to fail occasionally, whereas software, eventually, would become bug-free and thus cease to fail. The programmers [software engineers, we would now say] present did not share his confidence. There was too much unpredictable concurrency of program execution in the system to allow the 'provably correct' testing which was starting to be fashionable for batch processing. In any case, the operational and contractual requirements, of greater than 99.9%, 24x7 system availability, required a system design which would recover rapidly and completely from a system 'crash', however caused - within 30 seconds. To this end, the duplication and re-configurability of hardware has already been described - see 'Hardware' above.

This hardware redundancy needed to be complemented by bespoke software features, which, to varying degrees, were programmed into all parts of the application. Embracing them all was SYCOM, an evolving set of operator commands allowing complete control of configuration and recovery, down to specific groups of agent set terminals or individual SITA network circuits. 'Complete' recovery meant that, for example, all interrupted multi-step terminal transactions would resume from the last successfully completed step, using drum- stored transaction journals. It was many years later that 'standard' operating systems began to offer this degree of resilience, at a time when just re-starting the OS took minutes, anyway.

Usually, crash recovery would involve manual swap-over of online/offline peripheral sets between the two Central Processor Units (CPUs) and restart of each in its new mode. Each would determine its mode and peripheral population by a process known as 'grope'. Normally, the online CPU would have duplicated files on duplicated drum subsystems. If, however, a drum or subsystem were unavailable, the software would adapt. When a missing unit was restored, its file contents would automatically be brought up-to-date - concurrently with the ongoing online operation of the system. This feature also was not 'standard' for 15-20 years, except on a few premium-priced 'non-stop' system products.

For additional resilience, and for fallback in the case of cutover attempts, flight inventory files were backed up hourly onto magnetic tape, and PNR files daily - concurrently with the on-line TP operation, naturally. If a duplicate drum file subsystem was temporarily unavailable, then, following restoration, it would be reloaded from magnetic tape and then brought up to consistency against the surviving copy - concurrently with ongoing TP, naturally.

Design for Performance

A contractual requirement was that over 90% of interactive transaction steps had a response time of less than 2 seconds. It was recognised that most of the existence time in the system of a sub-transaction would be determined by drum accesses, not processing. Such accesses would be, for example, to read and write [normally twice] file records, update transaction journals [for recovery] and retrieve necessary program elements. Moreover, most drum transfers were small, and transfer speeds quite fast.

Most file accesses on Beacon, as in most TP systems, were reads. Normally, read access requests were spread evenly between duplexed subsystems. Writes, of course, had to be made to both.

CONTORTS gave drum subsystem interrupts highest priority, and special efforts were made, even hardware modifications, to reduce interrupts overall and to minimise interrupt processing under interrupt-lockout. All of CONTORTS, which occupied about 12k words, and all the online reservations application were memory-resident, to reduce accesses for overlays. All of this code was 're- entrant', to allow concurrent processing of multiple transactions; the purpose of this was, at peak times, to ensure drum subsystems never had to wait for the next access request. Files on moving head drums were placed so as to minimise head movements.

The passenger files, incidentally, consisted of the indices and the PNRs - passenger name records. They were arranged as what we later learned were 'inverted files', or even a specialised form of relational database. If a PNR involved, say, four flights, then it would be linked to four indices. Indices and related records were linked together in both directions, and each contained minimal data from the other to allow a degree of re-construction in the event of corruption - a feature which was to prove invaluable.

Instrumentation

Every aspect of BEACON's operation which might affect performance or reliability was counted or timed, so that performance was known and could be tuned. For example, not only were drum accesses counted, but also their duration. Transaction existence times were important in determining the optimum level of transaction concurrency.

Interline Message Handling

When Beacon Phase 2 cut over, including handling and generation of interline reservations messages through the SITA telegraphic message network, as it then was, about 60% of BEA's reservations came by this means, i.e. from other airlines, and only two or three airline systems had reached that stage. Thus most messages were keyboarded on tele-printers, by airline or agency staff, following, approximately, the AIRIMP message protocols. The SITA switching centres were then also not computerised, and could not store the [typically 75 bps, CCITT2-coded] messages. So if a reservations system could not receive incoming messages, business was lost.

Beacon had 16 input and four output simplex circuits. Each input circuit terminated in a Siemens receiver-transmitter, which punched and interpreted the messages on paper tape, which was normally then read directly, under system control, into the system. If and while an enabling signal was not received from the system, between each message, for any or all circuits, loops of paper tape would accumulate, preserving the messages for later processing. Messages were edited, then stored on drum queues for processing.

Input Error Correction

Experience with other early reservations systems showed that conventional editing of interline messages for AIRIMP format conformity produced about 70% rejects, which would then require human intervention; not good for business or productivity. GEDIT was an interpretive editing meta-language, which corrected and prepared queued messages for processing and response. As usual, a compromise was reached between efficiency and hardware resource use. Beacon settled for a rejection rate of about 7%, an achievement greeted with disbelief by visiting executives from other airlines.

Interactive terminal handling

Several hundred Beacon agent set terminals were clustered under the control of buffer processors, small, early minicomputers, which sent and received complete messages to the online mainframe computer. This cluster controller approach, reducing mainframe connectivity, data errors, processing and, above all, interrupts, became a standardised product line about ten years later, e.g. with ICL's 7502, 7503. Even today, many multi-user minicomputers are still directly connecting dumb terminals, on which any keystroke, unbelievably, causes a central processor interrupt.

Name Searches

A Passenger Name Record (PNR) held details of one or more passengers travelling together on one or more flight/dates on one or more airlines, with specific details, e.g. diet or disability, contacts etc. It could be accessed, most efficiently, by a reference number, or else by any name in the party and any flight/date in the itinerary. Hence the 'relational database' structure, as we later learned to call it.

Any call centre, and airline reservations were the first, has a problem with names. Beacon could, if necessary, search a flight/date's passenger list for best phonetic matches [e.g. Knight, Night, Nite, Nought], by character- sequence matching, or, most tediously, list all the flight/date's passenger names alphabetically.

Postscript

And what happened next? Beacon, and Boadicea, added further applications to their systems, until, in the early 1970s, Government decreed that the two national airlines should merge, creating British Airways. This raised the question about future IT strategy. Beacon won the technical, price-performance arguments, Boadicea won the political debate, in 1973. [That is one version, anyway!] Beacon was phased out. One factor was that UNIVAC abandoned out the 490 Series, which meant that a UNIVAC-based joint system [the 1110 was proposed] would not be program-compatible with either Beacon or Boadicea.

Meanwhile, a global war had developed between IBM and UNIVAC for leadership in the air transport industry, one of the few markets where IBM's dominance in mainframe-based systems was seriously challenged. To a degree, that rivalry lingers still.

In 1987, four leading UNIVAC airline users founded AMADEUS, a business dedicated to providing airline IT services, starting with reservations, to multiple airline users. This concept had its origins in the mid-1960s - too soon. AMADEUS commenced service in 1992, for airlines and also, crucially, for travel agents. It now claims about 150 airline customers, from British Airways and Qantas to Air Congo and Air Montenegro, some 400,000 'on-line terminals', 2,000 'IT experts', network centres near Munich, Sydney and Miami, 99.9% availability and some 12,000 messages per second. It claims market leadership. So, probably, does GALILEO, a rival service founded at about the same time by other European airlines, notably British Airways. Both are seriously challenged by the global expansion of SABRE, which grew out of the pioneering IBM system for American Airlines, and became the most profitable part of the business.

Beacon had a legacy, of questionable value, for ICL, in its efforts to address transaction processing convincingly. In the late 1960s, a group of Beacon veterans formed part of the System D Project, a TP-orientated operating system anticipating the advent of ICL's 2900 mainframe range. It came to nothing. In 1972, following the appointment of ex-UNIVAC Geoff Cross as ICL's Managing Director, a second, larger wave of Beacon veterans arrived including Ed Mack, who became Director of ICL's Product Development Group. VME/B was already established as the multi-purpose operating system for 2900 Series. Mack established a new TP operating system project, VME/K, considerably staffed by people with UNIVAC airline systems experience. For numerous and varied reasons, beyond the author's knowledge or competence to evaluate, VME/K became not a success, but a trauma. VME/K was finally terminated in 1981 by newly-installed Managing Director, Robb Wilmot.

Article Editor's note: This article is an abridged version of a talk given by the author to the Society at the Science Museum in February 2008. Michael Knight can be contacted at michaelknight242@tiscali.co.uk.

Footnotes and 'justified formatting' inserted by Lowell A. Benson, *VIP CLUB* webmaster, December 2008.