My Corner of the Time-Sharing Innovation World

Bo Lewendal

Editor: David Walden

As with many people, the path of my career in computing had unexpected changes of direction, in my case relating to the time I entered computing and my national background. I was fortunate to have a small role in spreading the then new technology of computer time-sharing systems.

My parents were from Sweden and I was born there. In 1949, when I was one year old, our family moved to California, where my father studied for three years for a master's degree at the University of California, Berkeley. Then we returned to Sweden, living in Stockholm for three years.

However, my parents had loved California, and when I was seven years old, we moved to New York City, where my father worked for the Swedish Chamber of Commerce. New York was planned as a stepping-stone back to California, and three years later we moved to San Francisco, where my father again worked for the Swedish Chamber of Commerce.

I had learned English in New York City, and I finished the rest of my schooling through high school in San Francisco.

Out of high school, I went to UC Berkeley, planning to be a physics major. A year or so into my studies at Berkeley, in May 1968, I got a part-time job as an electronics technician as part of a project at the university. Electronics had been a hobby of mine during my youth, and as a teenager, I worked at Lafayette Electronics in San Francisco doing repair work, first involving tubes and then transistor systems.

Project Genie

The project I worked on at UC Berkeley was Project Genie,[1] funded by J.C.R. Licklider of the Department of Defense Advanced Research Projects Agency (ARPA), an early time-sharing system development that influenced the evolution of time-sharing system innovation. Project Genie started with a Scientific Data Systems (SDS) 930 computer and modified it to be a platform on which a time-sharing system could be developed and run. SDS (which later became Xerox Data Systems) sold the resulting system as the SDS 940.[2]

My boss was the now famous Chuck Thacker, recipient of the 2009 ACM Turing Award. (Several other now famous computing innovators were also members of the project, and I'll list some of them later in my story). One of the projects I was assigned to was to take a magnetic core memory and thread wires through the cores in such a way that it became a read-only memory. At the time, I didn't know what the read-only memory

was to be used for, but I later learned that it was a prototype to be used for part of the development of a completely new time-sharing system (more about this later).

Although I was hired as an electronics technician, the leaders of the project also let me play with the time-sharing system, which I used from a Teletype terminal. Over time, I drifted into programming and my title was changed to "coder." In that role, one of the projects I worked on was coding for a PhD student working on recognizing letters printed by hand on a Rand tablet connected to a PDP-5 with an Evans & Sutherland vector display. The resulting software also provided the equivalent of a mouse in addition to text entry. This was done in assembly language. I also remember coding a memory test program, which did the original testing of the core memories used later in the BCC 500. In my spare time, I also wrote two chess programs with a GUI that used the handwriting recognition software and the vector display.

I don't remember the exact order of my getting involved with Project Genie and my later drift into programming and decision to focus my studies on computer science instead of physics. In any case, I began taking computer science courses. One of the introductory courses involved programming in Fortran and Algol on a CDC 6400. One of my side projects was figuring out how to call Fortran subroutines from an Algol program. I did this by reverse engineering the machine code produced by the Fortran compiler. I also remember taking a compiler course taught by another Project Genie developer, Peter Deutsch, a now legendary computer programmer.[3]

Berkeley Computer Corporation

After a while, several key people from Project Genie left the project to start Berkeley Computer Corporation (BCC) to develop a state-of-the-art time-sharing system capable of supporting many users. The design goal was to be able to support 500 simultaneous users, hence the name BCC 500 for the system. In March 1969, I was offered a half-time job at BCC as a programmer, which I happily accepted. The other half of my time was supposed to be spent finishing my bachelor's degree at the university, but my attention became focused on my work at BCC and away from my studies. Eventually, I dropped out of the university a year or so shy of a degree. As it turned out, I continued working at Project Genie as a coder for another six months until I started working full time for BCC.

Anecdotes

> ## The prototype BCC 500 computer was built and worked, but the first customer cancelled its order and funding run out.

I was just a lowly worker at both Project Genie and BCC, so I know little about things such as how BCC was funded or how much prior thinking about the BCC computer system design had gone on while the key people were still with Project Genie (or thinking about the Cal TSS,[4] of which I had no knowledge at the time). However, I do know that the read-only memory that I helped build as a technician was part of the process.

Some development may already have happened on the SDS 940. In any case, we initially used a 940 system for software development work at BCC. The company rented a SDS 940 at Shell in Emeryville, California, not far from BCC's location on 4th Street in Berkeley. The Shell machine was available for our use between 11 p.m. and 7 a.m., and my job was to go to Shell each evening to load our version of the time-sharing system and user files from a backup tape for our nighttime use and then back up our files to tape at the end of the night. In between, I often stayed at the Shell facility and did my programming on a local Teletype of the 940 system while the other BCC people were using the system via remote Teletype terminals, connected through 110 baud modems. Eventually, the BCC machine was working well enough that we no longer needed the 940.

One of my first programming jobs at BCC was coding and documentation for the macro assembler "MICRO" for the BCC 500 microcoded CPU.[5] To handle the many options possible with the BCC computer's long microprocessor instructions (90-bit instruction word), this assembler was pretty sophisticated.[6]

Another programming project was the development of an incremental backup system for the BCC computer's big rotating memory system using the magnetic tapes connected to an IBM 360 used as an I/O processor.[7] In addition, I did the initial debugging of the file system in the time-sharing monitor of the BCC-500. I was also responsible for the maintenance of the system software on the BCC 500.

Although I was a junior person in the company, I participated in design discussions and learned a great deal from the top ranking team leading the technical work at BCC, people such as Thacker and Deutsch as well as Butler Lampson (1992 ACM Turing Award), Wayne Lichtenberger, and other eminent computer people.[8] In addition to learning so much about operating system design, programming language implementation, and so on from this group of people, BCC also had occasional Friday wine tastings. From these, I learned about another domain that has been a source of fascination throughout my life.

The prototype BCC 500 computer was built and worked. However, as I understand what happened, the first customer cancelled its order and funding began to run out, I think as part of a general economic downturn. For a while, people were paid for half time (but mostly kept working full time). Then the company started laying off people, although for the last month my pay rate was increased to improve my bargaining position as I hunted for a new job. As the company shut down, a core group of senior people moved to Xerox PARC. I looked for a programming job elsewhere but couldn't find one. Perhaps my lack of a bachelor's degree was a problem or it was just that the economy was down overall. I also tried to again work as an electronics technician but was viewed as overqualified for that position, and no jobs in that domain were forthcoming.

Norsk Data

Eventually, I made contact with Norsk Data Elektronikk (ND) in Oslo, Norway. Ed Fiala, a friend and colleague from BCC, had a connection at ND through someone he had known in a prior job who now worked at ND (David Walden). My name was passed to Rolf Skår, who was a cofounder and software manager for the company.[9] Rolf had someone interview me by phone and check my references with a few key people from BCC, and I was offered a job. It was an interesting opportunity: I needed a job, and I thought that my Swedish would allow me to converse with the Norwegians in their own language

(although almost everyone in the company also spoke English) and moving to Norway could be a way to eventually return to my native country of Sweden. I joined ND on 3 March 1971.

As I remember, the main ND computer at the time, the Nord-1, was basically set up to work with files on paper tape. For instance, there was a text editor that read in a paper tape, let you edit the text, and then punched out the updated file. The Nord-1 also supported a real-time operating system called SINTRAN that was mostly used for process control applications. It was a more primitive development environment than I had been used to in Berkeley. Thus, I set out to build some new tools for the machine.

The first pieces of software I wrote for the machine were a package to do string manipulation and an implementation of the QED text editor that I had used in Berkeley (originally created by Deutsch and Lampson for the 940 system[10]). I also implemented a copy of the Runoff system from the 940 system to be able to do my own document production and not burden the ND typists. I don't remember how much of an official program there was for doing these projects. I did talk to Rolf about what I was doing, and he let me go ahead. Being a programmer himself, he perhaps saw the value of a programmer developing new tools to improve the software development environment.

On 14 June 1971, my coworker Torolf Paulsen and I proposed that ND create a time-sharing system for its Nord-1 machine. We presented two alternative system approaches, one using a multiprogramming system developed at Sentralinstituttet for industriell forskning (SI). However, this system would require that hardware paging be added to the Nord-1. The other approach was to build a simple time-sliced time-sharing system, which would require users to be completely swapped out of core memory when switching to the next user.[11–13]

My records show that I wrote a more detailed proposal for a simple time-sharing system on 5 July 1971.[14] Later in July, the company went on vacation for three weeks, as was the common Norwegian practice, which I had not known about. I asked if I could stay in the company offices during the vacation period to work on my ideas for a simple time-sharing system, and Rolf Skår gave me the go-ahead.

During the vacation period, I worked day and night. I implemented a disk-based file

> **Some of the Nord-1 time-sharing systems were sold to technical schools in Norway, often to run a Basic system implemented by an ND employee.**

system with QED interfaced to it. Then I wrote a basic time-sharing system with commands to access the paper tape punch and reader and the file system and a round-robin time-slice scheduler using software swapping among users. This system was working when the rest of the people in the company came back from vacation.

Obviously, I had benefitted from my knowledge of and experience with the Project Genie and BCC time-sharing systems in Berkeley. In addition to having an idea of the value of time-sharing on a minicomputer such as the Nord-1 (which at the time was not necessary for a minicomputer vendor competing in Europe), I also brought with me some Project Genie and BCC documentation.[15] While I often looked at this documentation, mostly I used those systems as a model and just decided what made sense to implement for the Nord-1. (Much of what was done at BCC was too sophisticated for the ND situation.)

Rolf Skår was impressed and happy with my time-sharing work for the Nord-1 and saw its market potential versus what other minicomputer vendors were selling. Thus, I continued improving the system, including a better scheduler and making the system run more efficiently. Some of these time-sharing systems were sold to technical schools in Norway, often to run a Basic system implemented by another ND employee (Jørgen Håberg).

The Nord-10 computer was developed by ND as a more powerful follow-on system to the Nord-1. I participated in some of the design decisions for this machine, and it included hardware to better support time-sharing among users. I kept improving the

Anecdotes

software to take advantage of the Nord-10 hardware and to improve the time-sharing system in other ways.[16] Also, after both the Nord-1 and Nord-10 time-sharing systems were working, I was sent along on many sales calls to provide technical information about the systems.[17]

In particular, when ND was bidding on a contract to provide 24 computers to CERN in Geneva, people from CERN visited ND and I demonstrated our time-sharing system.[18]

After ND won the CERN procurement, Rolf Skår, Trygve Matre (the creator of the SINTRAN operating system), and I drove to Geneva carrying the parts of the first system installation in our cars. When we arrived at the Swiss national border on Sunday, we had to wait until Monday when the customs office was open to enter the country. Once there, we had three weeks to get the system put together and working, which we did.

As the remaining ND computers were delivered and installed around the accelerator, I returned to CERN several more times for a few weeks to work on various parts of the system or to install things we had developed in Oslo.

One of my follow-on projects for CERN was to write the NODAL language interpreter. This was a language similar to Basic to which the CERN personnel could give commands relating to controlling and monitoring the accelerator. Instances of the interpreter existed in each Nord computer, and sets of instructions could be sent from one machine to another with interpretation at the destination machine in order to access that machine's interface with the accelerator equipment.

After Norsk Data

I left ND on 1 January 1974. Jørgen Håberg, who had also worked at ND, and I started a company in Halden, Norway, called Norsk Software Industri. Lars Monrad-Krohn, another of the three founders of ND was on our board of directors. We soon added Alf Martin Jensen and Alix Cordray as employees. From there, we kept working for CERN, continuing to improve NODAL, among other projects. One performance improvement to NODAL was a compiler of snippets of code, which had previously been only interpreted; the compiled snippets could then be sent to a particular computer for execution there. We also wrote assemblers, compilers, and interpreters for CERN and other customers. For instance, we wrote an RPG compiler for Modcomp and a PDP-11 Basic interpreter for

CERN so they had an implementation of Basic open to their own modifications. We only had a Nord-10 for our development work, and thus to develop a program for the PDP-11, we first had to emulate the PDP-11 on the Nord-10.

In 1977 I married and moved back to Oslo and left Norsk Software Industri. In Oslo, Alf Martin Jensen and I started a new company, Nittedata, also doing contract software work, such as macro-assemblers. Alix Cordray came along with us as an employee.

In 1983, the three of us closed Nittedata and, with another person, started a company named Hambo. Our goal was to build a CAD product for the architecture and construction industry. This CAD system did more than just allow the architectural design; it also produced construction drawings, parts lists, and so on. Rolf Skår was also involved in this endeavor as a shareholder and a member of our board of directors, and we got outside funding. In time, Hambo shares started trading on the Norwegian over the counter market. However, we could not find a sufficient market before the money ran out and the company was dissolved. We brought the technology to Orbit Creative Computing A/S, where we continued to work.

In 1987 my wife and I moved to California. In California I continued to work at a distance for the company in Norway. But eventually, that job ran out and I found a new job with Tektronix in Wilsonville, Oregon, and we moved to the Portland area. At Tektronix, I wrote drivers for color printers and worked on some of the embedded software for the printers. In 2006 I took early retirement from that division, which by then had become a division of Xerox. While doing that color printer work, I was listed on five patents. Back in my younger days, when I was more innovative, software wasn't patentable.

The year 2006 was the end of my programming career. Since then, I have worked with my wife who does English-to-Norwegian translation of documents. I help with proofreading, editing, and other aspects of her business. In 2012, we moved back to Oslo, where we continue to operate this small business. We look forward to becoming old enough shortly to collect our US Social Security pensions and to collect our Norwegian pensions a year later.

Thus, in a year or so I will have more time to reflect on my career in the computing world and the almost accidental steps that put me in touch with some of the technical giants

of our industry in Berkeley; then led to the years at Norsk Data, where bringing a little operating system technology that I happened to know to a small company contributed significantly to the company's astonishing growth; and finally allowed me to support myself and my family doing interesting work for additional decades.

References and Notes

1. "Project Genie," Wikipedia, http://en.wikipedia.org/wiki/Project_Genie.
2. B. Lampson, W. Lichtenberger, and M. Pirtle, "A User Machine in a Time-Sharing System," *Proc. IEEE*, vol. 54 no. 12, 1966, pp. 1766–1774; http://research.microsoft.com/en-us/um/people/blampson/02-UserMachine/WebPage.html.
3. P. Seibel, "Interview of L. Peter Deutsch," *Coders at Work: Reflections on the Craft of Programming*, Apress, 2009, chap. 11.
4. B.W. Lampson, "An Overview of CAL Time-Sharing System," Computer Center, Univ. of California, Berkeley, 1969; http://bitsavers.informatik.uni-stuttgart.de/pdf/univOfCalBerkeley/Cal_TSS_Overview_Oct69.pdf.
5. From a document and a webpage I have read on the Internet, I conclude that a different assembler named Micro was developed by some of my colleagues from BCC when they had moved on to Xerox PARC.
6. B. Lewendal, "MICRO Reference and User Manual," BBC, Sept. 1969; http://bitsavers.trailing-edge.com/pdf/bcc/originals/Memos/BCC_M-08.pdf.
7. B. Lewendal, "Disc Backup Subsystem," BBC, Jan. 1970; http://bitsavers.trailing-edge.com/pdf/bcc/originals/Specs/BCC_S-26.pdf
8. A list of some of the key people at BCC can be found on W. L. Harrison, "Supervisor's Manual," Oct. 1969, p. 2, (document p. 1), http://bitsavers.trailing-edge.com/pdf/bcc/originals/Admin/BCC_A-11.pdf. Not listed on that page is the now famous Charles Simonyi, who also was with BCC, having been recruited from where he was working as a programmer in the university computer center (according to his interview in S. Lammer, *Programmers at Work: Interviews with 19 Programmers Who Shaped the Computer Industry*, Tempus Books, 1989). I didn't work directly with Simonyi at BCC, but once I raced my Triumph motorcycle against his Triumph car for one city block.
9. D. Walden, "Interview of Rolf Skår," *IEEE Annals of the History of Computing*, vol. 35, no. 1, 2013, pp. 72–77.
10. "QED (text editor)," Wikipedia, http://en.wikipedia.org/wiki/QED_(text_editor).
11. B. Lewendal and T. Paulsen, "Time-Sharing System," June 1971, http://sintran.com/sintran/library/libsw/TSS-related-notes.pdf, pp. 1–2.
12. B. Lewendal, "NORD-1 Timesharing System," July 1971, http://sintran.com/sintran/library/libsw/NORD-Timesharing-System.pdf.
13. Shortly before my time at ND, David Walden had begun telling Rolf Skår that the Nord-1 needed a time-sharing system. In an undated memo to Rolf, Dave proposed a "simple time-slicer," providing a good description of how it would work.
14. B. Lewendal, "Proposal for a Simple Timesharing System," 5 July 1971, pp. 30–34; http://sintran.com/sintran/library/libsw/TSS-related-notes.pdf.
15. The 940 system has been described in the published literature (for example, see note 2). A great deal of BCC documentation is available at http://bitsavers.trailing-edge.com/pdf/bcc/.
16. B. Lewendal, "NORD-10 Design Goals," 1973; http://sintran.com/sintran/library/libsw/NORD-10-Design-Goals.pdf.
17. B. Lewendal, "TIMESHARING: What, Why and Whither," *ND NYTT*, no. 5, Sept. 1972, pp. 15–18; http://sintran.com/sintran/library/libsw/ND-Nytt-Wither-Timesharing.pdf.
18. ND's effort to bid on and win the CERN procurement has been described in the interview noted in Walden, "Interview of Rolf Skår," and in T. Olav Stein, "The Founding, Fantastic Growth, and Fast Decline of Norsk Data AS," *History of Nordic Computing 3, IFIP Advances in Information and Communication Technology*, vol. 350, Springer-Verlag, 2011, pp. 249–257.

Bo Lewendal *retired from Xerox in 2006, after working as a software engineer. After that he has worked, together with his wife, as a linguist. Contact him at* bo@lewendal.com.

cn *Selected CS articles and columns are also available for free at* http://ComputingNow.computer.org.