

611 Carroll St
Brooklyn NY 11215
718 809-0663
cmurthi@seeinggreen.net

13,500 words

The Poet, the Physicist, and the Immigrant

Birth of a Database

By Chandru Murthi

On the 4th of October, 1957, the Soviets launched the Sputnik satellite, a 23-inch-diameter double-hulled ball of gleaming metal which circled the earth every 96.2 minutes at about 18,000 mph. Sputnik had neither a computer nor any sensors to monitor conditions in space. Its only communication device was a 1-watt radio transmitter that broadcast signals to earth. Scientific data was inferred by the quality of the received radio signals as the satellite orbited.

Today, when your refrigerator sends you a shopping-list, you select your light levels by talking to your light bulbs, and the imminent arrival of autonomous cars, not having a computer in something as sophisticated as a satellite beggars the imagination.

Of course, a computer in that era would have meant a gigantic, room-sized system of enormous weight, with prohibitive power and cooling needs to boot. The aptly named *Colossus* computer, for example, originally used for code-breaking late in WW-II, had 2,400 vacuum tubes—devices that control electrical signals and were fundamental to computer operation—glowing like large incandescent Christmas tree

lights. There was also a paper tape reader the size of a king-size bed. This latter was a monumental novelty in itself, with tens of yards of ½"-wide paper tape fluttering on pulleys as it traveled at an extraordinary 50mph.

By 1955, replacement of tubes with solid-state transistors, invented in 1945, resulted in the first all-transistor computer the Harwell CADET (Transistor Electronic Digital Automatic Computer—backwards.) Transistors represented a triple-whammy compared to vacuum tubes—more powerful, using a fraction of the power, and with higher reliability.

A Start in Engineering

One June day in the early sixties, I faced the interview committee for admittance to the 5-year Engineering program at the Indian Institute of Technology, Madras. Progressing well, I thought, and relaxed. Then one of the interviewers suddenly asked me if I were the son of so-and-so? And I had a frisson of déjà-vu. My father had been denied entrance to an Engineering College four decades prior, because we were of the Brahmin caste. At that time, caste discrimination against Brahmins was rampant. This was a backlash against decades of Brahmins being top-dogs; they ran most everything, with strong support from the British. When the non-Brahmin castes came to power, they asserted their revenge by placing obstacles, academic and otherwise, in the Brahmins' path.

Fighting my natural contrarian urges, I said yes. Thankfully times had changed; false alarm; I was accepted.

The Super 'Copter

Fast forward to the Apollo lunar lander mission—1965. Computers had evolved and miniaturized, more so if one had the vast resources of a NASA. Apollo's Guidance Computer was about the size of a fat briefcase encased in protective metal, weighed 70 lbs., and drew 70 watts of power, about that of an incandescent light bulb. It sported a grand 36 thousand words of memory. Its execution speed was 1 million instructions a second—1 Megahertz.

Also in 1965, the US Army decided it needed its own, specialized attack aircraft, as until then they had to make do with customized versions of Air Force helicopters. Conceived under the Army's Advanced Aerial Fire Support System (AAFSS) program, the craft was eventually named the Cheyenne AH-56 Attack Helicopter. The Cheyenne was a radical aircraft with helicopter rotors, a short fixed-wing, and a forward propulsion system. These features would, it was hoped, marry the advantage of a helicopter with the flight speed of a conventional fixed-wing aircraft. In 1966, the Army awarded the main contract to Lockheed Corporation, located in Burbank, California, a suburb of Los Angeles.

One of the other companies involved in the Cheyenne development was TRW Systems, in Redondo Beach, also in Los Angeles County. TRW Systems had a long and surprisingly diverse resume, working in aerospace (satellites) building rockets, and had also been an early manufacturer of computers in the vacuum-tube days. They later branched into semi-conductors in the late 1950's. At the time of the Cheyenne's inception, TRW was building the descent engine for the Apollo lunar lander. The contract for the Cheyenne Helicopter project required developing a computerized inventory file system to manage the Cheyenne's thousands of parts (the term "database" did not exist yet.) Perhaps because of TRW's experience with manufacturing computers and complex software, this part of the contract was awarded to them.

The task of creating the file system software was given to two recent employees, a Donald B. Nelson, and a Richard A. Pick. TRW left the unknown dimensions of a "computerized file management system" to these relative newcomers to design and implement. Little did they know that this duo would proceed to create the first successful database system extant.

Nelson and Pick

Nelson, 40, had originally favored English literature and poetry (he once said of American literature "is there any?") studying at Stanford University. For a TRW hire, this might seem to be decidedly a curious talent, unrelated to either computer science, which was barely an

academic pursuit then, or information technology, another term not yet in use. Nelson, however, did have impeccable credentials.

He had been a protégé of Floyd George Steele, a pioneer in computers, and participated in the refinement of Steele's early and innovative Digital Difference Analyzer (later, with the addition of a Magnetic Automated Drum, known as MADIDDA, pronounced Mad Ida.) He had worked with Steele several years until the latter sold his small company CRC to Litton Industries. Steele left a few years later. Nelson, at loose ends, drifted for a while, selling all his possessions and moving to Samoa, well before such self-realizing trips were all the rage. Soon bored, he returned to the US, working as a silversmith in San Francisco while continuing his poetry study at Stanford. TRW eventually hired him in late '64. In the meanwhile, he also started developing a methodology of using "natural language" to communicate with computers.

Harry Emerson (a pseudonym,) one of Pick's later super-programmers, described Nelson as having a Buddha-like, friendly demeanor, and "...supremely confident in his ideas. Always expanded on [them,] and took listeners out of their comfort zone."

Pick, 27, born in Chicago, then Southern California bred, attended high school in suburban Covina, 22 miles East of Los Angeles. He graduated from UC Berkeley in 1961 with an MS in Physics, and had "completed coursework" for an MBA. He then worked as a programmer for the Mattel Toy Company, application-level work that he found quite boring. Looking for system-level work, he was hired by TRW in March '65.

Pick, almost 6' tall and always verging on slight overweight, had a piercing gaze that belied his otherwise laid-back manner and slightly slouched stance. When he focused on you, you would find it difficult not to be drawn in by his intensity. This ability to make his listener pay attention to him would serve him in good stead in his later business dealings. His easy (and in those pre-PC days, scatological) sense of humor and characteristic barking laugh helped, too. Pick also seemed singularly attractive to women. By then he had had two children. His wife Anne was from the wealthy class of Newport Beach, California, just south of Los Angeles.

Pick's chief programmer was Philip Earl, who would join TRW in early 1965. Earl was a graduate of UC Santa Barbara '61 with a degree in mathematics. He had worked at Northrup Aircraft as a computer trainee programming for an IBM mainframe before joining TRW Systems. In contrast with Pick, Earl was easy-going and laconic, with a dry sense of humor. His interests were varied, including photography and video, which he put to good use later. Surprisingly he would say later that he and Pick were never close, in spite of many years of working intensely together.

Computers and the English Language

In 2018, you may be reading this, particularly if you're under 40, on a hand-held device, a Kindle or a smart phone. The raw computing power of your device would have outstripped the Apollo Guidance Computer by a factor of 10,000 or so. Your child is on an iPhone, loading the latest video sent her by her friend. The iPhone's computer, if you have ponied up for the top of the line, is clicking away at 1,900 Megahertz and has 256 Gigabytes of memory, an almost 2,000 times increase in speed and 6 *million* times the memory storage of the Apollo Guidance Systems computer. You could say that an iPhone would have the power to guide a fleet of thousands of Apollo landers to the moon simultaneously.

Those are big multipliers, not easy to wrap your head around. 10,000 times, or as computer people prefer to call it, four orders of magnitude, means that your device is executing that many times at its internal raw speed, compared to the then recently released (1964) IBM 360 which TRW systems may have been using. Its speed was a mere 1.4 Megahertz. The base IBM 360 was priced at \$253,000, which translates to a cool two million in today's dollars, and a typical system would have been three or four times that price with required peripherals.

Now, why it still takes several minutes for your ultra-high-speed device to boot when shut down is a question for the ages.

Computers were by definition at the time, "number crunchers." The internal central processing unit (CPU) basically could add, subtract, and perform a few other arithmetic and logical operations. Nelson had

the advantage of not being particularly computer-centric, which in those times meant having a degree in Physics or Electrical Engineering and wearing pocket protectors as you discussed Gaussian Fields and other esoterica with the other men around. Few women applied, though interestingly enough, women then predominated in writing software, which the men considered secondary work—until it became prestigious and lucrative.

But Nelson was a highly intelligent and common-sensical person, and he had long decided that mere mathematical ability did not a computer limit. He set out to define a new style of “computer” which could manipulate strings of data—your name for instance, or a description like ‘Nut, hardened steel, 3/4 Thread,’ a vastly more useful ability, particularly if your agenda was control of a million parts for a helicopter. He proceeded to write a set of “flow charts,” or computer instructions, for this non-existent machine, instructions which were like nothing seen before. They were not mathematical computations, but more like “move this string of data from here to there.” Amazingly, Nelson’s flow-charts were detailed enough for a programmer to more-or-less translate them directly to computer instructions.

Within a few months, in March 1965, Nelson delivered his voluminous USER REQUIREMENTS SPECIFICATION. The TRW upper echelons found the document difficult to understand, written as it was in an academic style. Nelson articulated his methodology of using “natural language” in phrases like:

- ☐ The definition of “language” imposed on the system requires the accomodation [sic] of inputs stated in a natural language limited by a minimum of system rules and restrictions.
- ☐ Such a “natural language” requirement is assumed literally and not merely as the antonym of “machine language”.

Also unhelpful was Nelson’s renaming of standard computer jargon (“attribute” for “field”, “item-list” for “file”.) But he had expansive and revolutionary ideas, and the term *virtual*, as opposed to *real*, would feature strongly in them.

Note on Virtual Reality

Virtual, in the Miriam Webster Dictionary's quaint first definition, is: "being such in essence or effect though not formally recognized or admitted. Example: a virtual dictator." Let us admit that, in 2018, we are more likely to agree with this one: *Virtual* (adjective): not physically existing as such but made by software to appear to do so.

So our contemporary *Virtual Reality* is a computer simulation of conditions we perceive as if it *were* actual, realizable, "reality." And it's all thanks to the incredible speed of modern computers and the incredible programming abilities of young computer nerds.

Back in the '60s, people would not have fathomed the second definition of *virtual*, but in the computer world, *Virtual Memory* was a term just gaining ground, and of much importance in the era of simpler computers. A computer can operate only on data, or other information, in its memory. The disc drive, or possibly magnetic tape in those days, is where the vast majority of its information was stored. Data must move from the disc to the memory to be usable—no different than your not knowing a fact which you remember you have previously written down on page 17 of your Banking folder in your file cabinet. A quick grab and perusal of that page transfers the information to your brain—your "real" memory.

The issue was that hardware memory of computer systems was both extremely limited and extremely expensive. Extremely limited because the memory-addressing capability of those computers was limited. Think if you had a limit of 4 digits on a phone number instead of our current 10, you would be able to reach only 1,000 exclusive telephone users.

Extremely expensive because memory boards in the years '64-'74 consisted of thousands of "cores," little magnetic toroids (doughnut shaped circles.) Each represented one bit and was about the size of a pencil-point, attached to a board and sporting *three hand-strung* wires through each central aperture. Attempts to automate this procedure never came to fruition, and memory manufacture soon moved abroad to countries with cheaper labor, but boards were still pricy. We can imagine an army of women—and they were almost always women, many drawn

from the dress-maker trades for their superior hand-eye co-ordination—equipped with microscopes, carefully and tediously inserting a hair-thin wire through one core after another. Repeat several thousand times.

A memory board with, say, four thousand words therefore took a while to build, test and deploy. The Apollo Guidance Computer had only 36,000 words of memory, which, at the time, might have cost \$50,000 or more. Later, of course, solid-state memory chips changed that dramatically. The device you're reading this on may have *16 billion* words or so.

The Virtualization of Everything

Faced with these limitations, Nelson and Pick decided to bypass the entire problem of “real (hardware) memory,” and implemented a “virtual memory manager.” This controlled information as it moved in and out of real memory as required by the computer's operations.

It would be as if you were told, with no prior notice, that at the next meal you're cooking, you will have 24 guests. You go to your minuscule pantry and look for 20 potatoes, 15 onions and three pounds of butter, none of which are physically present when you open that tiny door. Luckily you have verbalized your needs, and quite soon, all the foodstuff appears magically. Since the pantry is so small, other, unneeded goods also magically leave to make room. It seems you have a co-operative grocery store next to your house which is both willing to send you what you want, re-stock what you don't, and the mechanism to perform the food transfer. Lucky you, you have both a virtual pantry and a virtual foodstuff manager.

The concept of a virtual memory manager was not new; its earliest incarnation was in the University of Manchester's *Atlas* computer of 1962. The Atlas had specialized, and expensive, hardware to perform the management. In the nascent database, Nelson and Pick programmed the memory management in *software*, which greatly simplified and reduced the cost of implementation. The disc drive was treated as if it were an extension of real memory. Pick, in a later overview of this system, wrote, “With the Pick system, users [i.e., programmers] need not concern

themselves with the memory requirements of programs, files, or reports.”

There was more. In addition to the memory-enhancing breakthrough, Nelson and Pick had also created a *Virtual Computer*: one that simulated their new-fangled instructions on a real one. *Simulation* is computer jargon meaning that the new instructions translate to one or, more typically, many instructions that can actually run on the computer. This of course makes each of the new instructions run several times slower than the actual instructions, but the power of each Nelson instruction more than compensated for the reduction in speed. It would be like using a large but slow van to transport vast amounts of goods instead of using a small fast car requiring many more trips.

The actual implementation of these instructions was left to Pick and his team to fathom. The genius of Nelson was conceptualization; that of Pick was implementation, a serendipitous combination indeed.

Talk to me—Alexa

We are in the golden age of human-computer communication. Perhaps that’s a dangerous thing to say, as the sheer rapidity of computer progress makes such statements soon outdated, and prognostication futile. But most of us would be well satisfied with our written Google, or verbal Alexa, interactions. Google, for the most part, even does a good job predicting what you are looking for.

Think pre-Google, pre-internet, pre-PC. You might never have seen a computer unless you were in the field, and almost certainly never demanded anything from one. If you worked in an office which had a computer, you’d have to ask the resident computer expert to get your data from the computer in human-readable form, which might have taken hours, even days. Amazingly, this disconnect still exists today—business computer databases are treated as fragile repositories of information, accessible only through approved and rigid channels. Google has not yet conquered the average corporate computer.

Back in 1965, the charter from the Army about a “file management system” included an ill-defined statement about ease of access to, and update of, the data. Since this type of interaction did not exist anywhere,

Nelson was free to interpret the concept as he willed. And he had grand ambitions, perhaps because he did not like to think in computer-programming terms. Trained in literature, the English language loomed large. Why not use it to query the database?

Nelson designed and flow-charted a free-form language (one without a lot of rules,) used to retrieve data from the file system and to display it in a more-or-less human readable format. Integral to the system, the data retrieval mechanism was fancifully named *GIRLS*...for Generalized Information Retrieval Language System. It was possibly the first ever easy interface to a computer. It was usable by non-computer-savvy people with practically no training, a task which normally would have required a highly experienced computer technician.

While the Google-spoiled amongst us may scoff at having to type something like *LIST THE INVENTORY.FILE WITH DESCRIPTION LIKE "Nut, hardened steel, 3/4 Thread" SHOW QUANTITY AND LOCATION*, this was quite revolutionary in 1968. Later, Nelson began work on an equivalent, English-sentence-like, ability to modify data in the database: *CHANGE SALARY FOR EMPLOYEE 2144 FROM 30,000 TO 35,000*.

Upon receiving a prototype of this software in 1969, the Army called it "...by far the finest generalized information management system in the country."

Pick thought very highly of the functionality of the system, too. He later said, "the system that Don and I developed enabled the Army to pay for the entire computer system that TRW had developed, several times over, by identifying Lockheed cost overruns."

Math vs. Logic

In 1969 I started an MS in Industrial Engineering at the University of California, Berkeley (Pick's alma mater.) I had graduated the year before from the Indian Institute of Technology, Madras with a bachelor's in Mechanical Engineering. I'd had several job offers but didn't really want to stay in India. Even at 21, I knew it was too rigid a climate for me. The thought of spending fifteen to twenty years to wind up as a mid-level

manager at IBM or Tata Industries was horrifying. My turning these offers down was considered scandalous—what were my parents thinking? Luckily I had, by Indian standards, quite non-intrusive parents. Many of my fellow-students were continuing their studies in America—a brave few in Germany, as IIT Madras had a close connection with that country. I chose, by default, America. I was well-steeped in Western music, movies and *morés*, and my entire education had been in English. There would be no culture shock.

My first exposure to computers—IIT did not have any, or any that I was allowed to see—was with a batch-style hulk which took inputs from a card reader. The reader was another behemoth that ingested envelope-sized cards on which information had been punched, much like current ballot cards. You would find a line of students clutching their card decks protectively, waiting their turn—and if, like me, you'd forgotten to turn on the sequencer on the card punch, you didn't want to drop your deck. That would mean spending the next few hours re-sorting the un-sequenced cards.

Later I moved to a relatively new Digital Equipment Corporation's PDP-11 computer and discovered, much to my surprise, that programming came easy to me. Surprise, because I was abysmal at higher math. I had barely passed my second year math final at IIT by the expediency of writing an elaborate math proof from both ends towards the middle, leaving out a crucial (and, for me, unknown,) section. I got full credit from an inattentive grader. I soon realized that strong logical ability was more important than math for the type of statistical and systems programming I would deal with.

Pick Decamps

The Air Force had been never too happy about the Army's obtaining a most sophisticated helicopter, and had been vociferous about its objections to the development program. By 1971 this pressure, as well as a fatality in the 1969 crash of one of the Cheyenne prototypes (the third to crash,) resulted in the projected cancellation of the production contract, leaving orphaned the file management system, virtual machine and all.

TRW seemed to think this development effort had little chance of commercial potential, so Pick decided to leave TRW. Pick and his programmers joined a small Los Angeles software company, General Analytics, where he continued development of what he would soon call the "Pick System."

He had the boldness to take with him several software tapes (in those days, fragile magnetized tape on large 12" reels.) Much later, he would say that "the reason I didn't feel badly about ripping off TRW was because they didn't [expletive] own it." Nelson did, he thought, because Nelson had formulated much of the underpinnings before he had joined TRW. In saying this, he was both self-serving and sincere; much later, he would prove a savior to Nelson when the latter hit hard times.

Pick also had the foresight to send in the paperwork to the Army to request a copy of the same software, which TRW had left, as was then the norm, in the public domain. This would take several months to obtain, so the TRW tapes gave him a head start on continuing development. The request to the Army proved a winning ploy as, a couple of years later, TRW initiated a suit against Pick and others, claiming theft of trade secrets. Pick's successful defense was the paperwork trail with the US Army and the physical tapes they had sent him.

Nelson declined to join Pick at General Analytics in the continuing GIRLS adventure. He worked for TRW for a while. Development of the database was transferred to TRW LSI Products, the software arm of TRW, who attempted to market it but failed.

General Analytics

I graduated in late 1970 from UC Berkeley with an MS in Industrial Engineering/Operations Research. OR, though a mathematical discipline, concentrated on statistical-and-logical areas of math, enough that I could cope even with my math-challenges. I had been desperately looking for a job since graduation, moving from Berkeley to Los Angeles. Unfortunately, much work in my line was with companies like Lockheed and TRW Systems, which required a security clearance available only to citizens, and thus not to me.

In another odd hire, General Analytics offered me a job to work on a contract for CBS Television which required mathematical modeling, a forté of Operations Research methodology. Why they thought a callow graduate with no experience could bring home a major contract was a mystery to me. My boss Warren Morrison seemed to be supremely confident in my abilities, and told me to “just write a proposal,” after giving me reams of paperwork on the issue. Needless to say, General Analytics did not get the contract after my lame attempt. I expected to be fired, but Pick took me into his group, mainly because of my knowledge of FORTRAN, a scientific language needed in one of his projects.

The programmers at General Analytics had just gotten new sleek and sculptural IBM Selectric computer terminals—and in non-beige colors! These were the print equivalent of today’s “monitor,” and were used to communicate with the computer, a vast improvement over the pre-WW-II era Teletype machines which were mostly the norm. The Selectrics ran at twice the print rate as the Teletypes—all of 17 characters per second, a speed that would still take it *over a minute* to print this page. I was to design “terminal driver software” for them in FORTRAN. At that time I had never heard of the term “driver,” nor seen a real “terminal”.

Pick proved to be an easy-going mentor, and tolerated my inexperience. We quickly developed a sound and lasting relationship, as I was good at interpreting his many randomly sketched and sometimes impractical ideas. We shared a June birthday, eight years apart, and he would crack that the reason we got along well was that “our biorhythms were totally in synch.” Californians really did talk like that then.

Pick also looked ahead: the power he saw in the database software could democratize the use of computers, not with mainframes but with the newly emerging, smaller, less expensive “minicomputers.”

The Advent of the Minicomputer

If you have read *The Soul of a New Machine*, (Tracy Kidder, '81) about the development of a minicomputer at Data General Corporation in the late 1970's, you would have learned a lot about both the internals of

minicomputers and the business decisions that spawned that genre. Back in 1970, there were many companies who had realized that, while the \$3+ million computers like the IBM 360 and the Xerox Sigma 7 were incredible performers, they would never be the mass market computer of choice.

Digital Equipment Corporation (DEC) had introduced the PDP-1, its first minicomputer in 1961, then the PDP-8 in 1965, which was priced at \$18,000 (\$140,000 today.) The PDP-8 proved to be a runaway success, selling 40,000 units eventually (this might translate in modern terms as iPhone-like.) For the first time, an average company could afford to computerize their operations. Not that this was a simple task; even the best minicomputers were limited in memory, speed and disc drive capacity.

One of the smaller players in the minicomputer world was Microdata Corporation, serendipitously located in Irvine, about 50 miles south of Los Angeles. Pick investigated Microdata's product and found it singularly appropriate to his needs. The Microdata-800 minicomputer was not only reasonably priced for its capacity (not at the top of the computer whiz scale, but a solid performer,) but had "firmware." This was a term coined by computer scientist Ascher Opler in 1965. It referred to programs that could be loaded on to a small, high-speed computer chip—in essence, "software" converted to "hardware," thus greatly reducing development time, and enhancing speed and reliability.

In current times, of course, firmware, and the dreaded "firmware driver update needed" message, are ubiquitous—from washing machines to modems to PCs themselves.

What got Pick excited was that the Microdata minicomputer was totally firmware controlled: it had no innate "mind" of its own.

Imagine if you will, finding a French Language Upgrade to Brain (FLUB,) which could be "uploaded" into the language center of your brain. You would then speak perfect French. You might miss not speaking English, but you should have paid up for the Multi-FLUB version which retains any pre-existing language abilities instead of overriding them (the computer jargon for this ability is multiplexing.) And what if the language-policing Académie Française, as they are wont

to do, imposes a fresh set of neologistic French words to replace the English ones used to date? FLUB, like firmware, is flexible. Just ask for the FLUB Rev 2.6 to be uploaded to your brain. Voilà! Never more say "subprime," say now "prix hypothécaire à risqué".

Changing the firmware in the Microdata-800 amounted to a computer brain-transplant—it would obey a completely new set of commands. Perfect for the Pick database system, written in Nelson's peculiar language. As Pick would say, if you can't bring the language to the computer, bring the computer to the language instead.

Morrison's Garage Band

General Analytics declared bankruptcy shortly after I started with Pick, and he and his system were once again orphaned.

Warren Morrison, Pick's boss at General Analytics, was a genial, perhaps over-optimistic soul who completely believed in the product Pick was developing. He offered to find funding for continuing development, and the use of the garage in his house in Los Angeles to work in while things were shaping up.

Pick, using his considerable persuasive skills, got Microdata to loan him a computer system, and for about six months, Pick, Earl, the other two programmers from TRW, and I were ensconced in a two-car garage with the Microdata-800 minicomputer and a great deal of enthusiasm.

The 800 came complete with the obligatory paper-tape reader, thankfully considerably shrunk from the Colossus computer's bed-sized version. This was the only way you could boot the machine from every power-off. And a magnetic tape drive. No expensive Selectrics for communication this time, we had to make do with a clattering Teletype machine. The staccato rattling did not improve the disposition of the many Siamese cats whose domain we had invaded.

A 5-megabyte disc drive was delivered also. This was a device twice the size of a large modern laser printer, making ominous clicking and chattering sounds as it sought data from a set of 14" magnetized platters, capable of moving the entire computer system cabinet if its

castors were not locked. A far cry from that 512-*gigabyte* thumb drive you're holding: a mere 100,000 times increase in capacity, with no moving parts to fail, either.

Soon it became apparent that Morrison could not obtain the necessary funding. Pick and Morrison agreed to split; two programmers elected to stay with Morrison; Earl and I decided to stay with Pick.

What is this thing called "Software?"

A few months after the bankruptcy, I received a deposition notice from the bankruptcy lawyers. Not paying much attention, I arrived at the lawyer's office with my girlfriend, Susan Spencer (we had been shopping nearby.) To my surprise, we were ushered in together. I just introduced her by her name and no questions ensued.

The matter the lawyers asked me about centered on the \$20,000 per month that a computer service company charged our five-person group at General Analytics for "computer timesharing charges." Timesharing of a computer system—the ability to share its resources and computing power amongst multiple simultaneous users—was not uncommon at the time, and an industry had developed in response to the exorbitant million+ dollar cost of computers. Timesharing still required a powerful mainframe computer, and typically, a host of special-purpose software, and sometimes hardware, that could "talk" to, and manage, the various connected users, but the cost to each user was tolerable.

The lawyers were monumentally ignorant of computers, software, and timesharing. After trying to explain what we were doing, they still didn't understand "where the money was going and why." After much back-and-forth, one triumphantly produced a memo I'd written.

A little background: We were using those colorful IBM Selectric typewriters to communicate with the remote system. The timesharing service offered, helpfully, an option whereby you could make the Selectric periodically type a single letter as the computer worked on your program. Each signal indicated that you had used one dollar of computer time. I had written a memo detailing the steps to make this

happen. All of us set the letter to a "\$", thereby seeing a string of \$'s appear when we submitted a lengthy or complex program. We thought it quite humorous watching the \$'s march steadily ahead.

The lawyers did not. Obviously this was a smoking gun of sorts. Why did I tell the others how to print \$ signs? Pointing out that it made no difference to the amount the timeshare company charged us seemed to do no good. I began to realize I might be in some unspecified trouble. During a pause, one of the lawyers turned to Spencer and said, "Perhaps you should advise your client to take this matter more seriously," to which she said,, barely concealing her mirth, "I'm not his lawyer." "Who are you then?" "Aah—girlfriend?"

Susan was politely asked to leave the room. They moved on: what had we produced to justify such expenditure? Software. Where is it? On the save tapes. Which are what exactly? And so on. Eventually they gave up and let me leave with a dark threat that I might be hearing from them again. Thankfully, I never did. I suppose someone more articulate than me had explained the software development business to their satisfaction.

Irvine, California

Richard Pick lived in Newport Beach, Orange County, about 50 miles south of Los Angeles. In 1972, when he decided to move his team of engineers there, he found an eponymously named complex called Sky Park Circle in the "village" of Irvine. It was, indeed, a circular two-lane road with low slung, industrial grade buildings on either border. Singularly devoid of a scrap of grass or greenery, it matched the ethos of the planned community of Irvine, where the concept of separation of uses, a planning term that meant isolation of commercial, industrial and residential areas, was carried to an extreme. Trees and greenery would have been considered superfluous and possibly left-wingish for an industrial zone in the then rabidly Republican atmosphere of Orange County. It was home to both the Ayn Rand Institute and the notoriously red-baiting John Birch Society (fittingly, the JBS moved in 1989 from the rapidly diversifying Orange County to Appleton, WI, birthplace of Senator Joe McCarthy.)

In spite of its name, the “village” of Irvine’s environment was cartoonishly car-centric. There was a grid of extra-wide superhighway-like roads spanning the central business district and reaching, tentacle-like, into the all-single-family residential areas a few miles in every direction. To get lunch at the restaurant across the street, whose welcoming sign you could clearly read when you stood at the outer edge of Sky Park Circle, you would have had to cross eight lanes of traffic. Technically, California road rules give pedestrians complete right-of-way, but you would be brave indeed to interfere with the progress of cars whizzing by at 50+ mph. You could decide to walk back and forth via the nearest intersection, possibly a quarter-of-a-mile away. You probably drove. Parking was never an issue.

There were few available distractions. You would have been hard-pressed to find any sort of public facility, park, museum or small hang-out coffee shop nearby (Starbucks was just a gleam in the eyes of its founders.) Productivity might or might not have been improved in this stultifying atmosphere, but that was the prevailing model for planned communities in the ‘70s. In any event, the team at Pick was youthful, energetic and could create their own distractions in the California post-hippie era.

Money to run the company was the sticking point.

Venture Capitalism

In 2018, you may envy your neighbor’s n’er-do-well 21 year old son inventing the app-to-end-all-apps, and of venture capitalists fighting each other off to throw a few millions into this wondrous marketing opportunity.

As it turned out, VC funding was difficult for a fledging company in 1970. No angels made themselves known, and the venture capitalists who would today have salivated at this radical concept did not yet exist. Pick might not, of course, have been happy about the prospect of diluting his sole interest in his new technology. He might have remembered that Nelson’s mentor, Floyd Steele, had also created an iconoclastic technology (MADDIDA,) which he nurtured for a while. He

had sold his company to another to get much-needed capital. Within a few years, Steele left “in disgust” because he couldn’t stand the intrusions on his work.

Pick once said “making money is an indication that you are doing something right.” And, “If you want to make your ideas happen, you must do it with your own money.” He identified one of his childhood idols as Disney’s Scrooge McDuck. He said he envied the way McDuck “was sitting on top of a big pile of money and used his riches to go off and do lots of high concept endeavors such as building cities in the Amazon.” This led him, in his words, to be “more or less financially independent by the age of twelve.” As a child he once hung out holding open doors of a restaurant to let patrons enter and exit, and was surprised when they started tipping him. His mother stopped him, not amused.

So losing control via investors was a thorny issue for him. But he needed funding. He had a house in Newport Beach which his wife’s parents had bought for the couple on their marriage several years before. Mortgaging this house for \$50,000 provided the initial funds for Pick & Associates. There was little to go around after capital expenditure and leasing; Pick promised all the junior employees compensation deferred to when the company had more money to distribute. Most of the employees subsisted discreetly on unemployment benefits. I was a non-citizen immigrant ineligible for this benefit, so was paid by Pick the princely sum of \$80/week. Luckily, my girlfriend Susan’s steadier job helped us live a decent lifestyle.

Pick & Associates was incorporated in mid-72. To Pick’s credit, he distributed stock in his company to all his employees, with no vesting or other encumbrances.

Pick Systems on its Feet

We moved into a two-story structure with several offices in front and a large warehouse like space in the back. Lacking money to outfit the place in normal fashion, Pick got a team of handymen to improvise desks and bookcases; plywood and bright paint brought a festive atmosphere to the programmers’ offices. Pick’s programmers were an eclectic group. Of the

original four, only two, Phil Earl and I, had joined him in Irvine, the commute or relocation being unattractive. He had meanwhile hired others.

Besides working on the Microdata computer, Pick was also using a large mainframe computer—a Xerox Sigma 7—at the nearby University of California, Irvine, for additional development. He would dread the monthly bills, which he regularly sweet-talked into postponement. While on campus, he ran into a Kenneth Simms, a recent UC Irvine graduate. Pick was captivated by Simm's intelligence and computer knowledge, if not his laconic demeanor, and immediately hired him.

Simm's main loves, besides computers in general, seemed to be playing and improving the then primitive Star Trek interactive computer game, developed by UC Irvine student Mike Mayfield and others that year. His other was consuming a can of diet soda every 20 minutes or so. His office eventually had many Star Trek posters on the walls, always a case of two of unopened soda, and 50 cans of empties. The sodas did nothing for Pick, but Simms' yen for Star Trek would soon lead to an unexpected bonus for the company.

Harry Emerson was another hire. Recommended by a friend, he had a degree in mathematics and a PhD in US Government, 1972, from Harvard. With minimal programming experience, he talked himself into a job with us, and proved to be an outstanding programmer. Unlike Simms he had a completely practical and no-nonsense approach to issues.

The balance of views, practical and theoretical, amongst us original programmers was extremely valuable when we hashed out our designs. Unique as the Pick system was, we did not have too much history to rely on, so thinking outside of the box was the norm.

The office atmosphere was certainly not formal. This was California in the just-post-hippie days. Inappropriate humor abounded, egged on by Pick himself. Our requisite Southern-California-blond secretary, the only woman in the office, was quite unfazed. Substance use was common. Many beers were openly drunk. Pick's choice was the yellow-can Coor's Banquet Beer. Pot was consumed a little more

discreetly, perhaps because it was illegal and Orange County was known for its crackdowns on “demon pot.”

Except on Fridays. There was an informal end-of-work timeout at about 5pm, and pot was certainly on the agenda. Harder stuff was consumed upstairs. Wives and girlfriends often showed up, and the speakers screamed *Surfing USA*. Pick himself was in full form, all ribald jokes and amusing anecdotes as his staff surrounded him like courtiers.

Frankenstein Wakes

With Pick and me in charge of developing the firmware for the Microdata computer, the others were working on software utilities and functions on the Sigma 7 emulator at UC Irvine. Pick was often off the wall in his design and demeanor. A typical Monday morning would have him storming in at 11am or so, and calling the programmers together.

“We need a super-efficient sort algorithm,” he said on one memorable day.

At that time, when you requested data from the Pick database, it would display in whatever order the file system had stored it. You could not display it, say, ordered by a person’s last name. Of course, in real life, this was unacceptable; we need to see data the way we want, not in some arbitrary way the machine stores it. However, ordering data, or using a *sorting algorithm* in computer jargon, is a very complex task when faced with thousands of pieces of data, and is also very computation-intensive.

“Here’s a fantastic algorithm I found, and, here’s the flow-charts for it,” Pick said as he threw sheets of paper around, discolored with streaks of ash—clearly from his marijuana-or-worse-soaked weekend labors—then walked to the blackboard and began madly scrawling, disclaiming on how the sort would be implemented.

Most of us were baffled, not so much by the arcanity of the method but because we were not trained to quickly “read” algorithms.

Simms jumped in, "Oh that's a n-way poly-phase sort-merge," he said authoritatively, taking several swigs off his diet Coke.

"Of course," some of muttered, not having a clue.

"Very efficient. Has anyone ever actually implanted this version? Where did you get that, Dick?"

"From Knuth of course, with a few of my mods. Best thing since sliced bread. I promise you, this will be fastest sort ever produced, we'll run rings around the Sigma-7!" A machine with a execution speed more than 10 times ours.

He spent a long time describing his breakthrough, which was the internal connections he'd made to maximize use of the Pick computer's string (as opposed to numerical) instructions. It did indeed make the sort efficient, masking the innate slowness of the Pick system. Heated discussion followed

Pick finally called on one unlucky soul. "Just do it. Follow the flow charts. It'll take a week at most!"

There was the usual bright idea hiding in the chaff of Pick's overblown verbiage, which could, and almost always would, be implemented. Not in a week, when only a sputtering and buggy version would placate Pick's impatience, but a month or two later the sort would be clicking away reliably.

"SNU loops! Who's got ideas for SNU's?" he demanded another day.

SNU stood for Select Next User. In a multi-user, timeshared computer system, one "user" (process or application,) runs for a while, but has to give up control to others. The question is: when? Some apps can run forever in computation-intensive cycles; some need data from the disk drive and have to stop and wait; others need attention from, say, a user's keyboard. Rotating amongst these diverse needs is no easy task; PhD theses could have been written about the strategies.

Pick, having absorbed various algorithms, wanted a consensus.

“Let each app run until it needs something external, then relinquish control,” suggested the democratic-ideal contingent, including me.

“We have an OS don’t we?” countered Pick, “what’s it for if it doesn’t control this shit? And what about prioritizing...we need to set some tasks to be more important than others.”

The “operating system,” or OS, is the set of programs integral to the computer which controls files, displays, and other crucial tasks—on a current Mac, it might be called OS X El Capitan; on a Windows PC, you might be running Windows 11.

“Give Ken the ability to set his programs high-priority and it means *Star Trek* will totally hog the machine,” I said, noting that Simms was suddenly paying more attention.

Pick and Simms preferred the authoritarian approach. The OS would intrude to see what each app was up to, and stop it when it decided enough was enough: In the end, given that the more rigorous the monitoring, the more it would slow the system down, we settled on a *laissez-faire* attitude, with only a two-level low-or-high priority.

Much of what we implemented was not rigorously tested for correctness; we relied on our gut-level instincts, which worked surprisingly well. This was mostly because of our lack of resources, but it also fit in with our prevailing ethos. In looking back, I wonder if being from diverse Californian backgrounds had something to do with it.

Within a year, the first re-brained Microdata 800 was powered up, booted up with a few flicks of the front console switches (no more paper tape by then,) and ready to accept its first GIRLS command, as indicated by the coy printout of a “:” on the IBM Selectric. Display monitors were still too expensive for us.

The group, consisting of Pick, Earl, Simms, Emerson, our hardware hire Byron Seagraves and me, crowded around the computer. Had Nelson been with us, I’m sure the poet in him would’ve come up with a suitably memorable first GIRLS statement to query the fledging machine.

As it was, I recall someone typed in an unimaginative *COUNT THE EMPLOYEES*. A slight and nerve-racking pause; then the Selectric chattered out the correct answer; and we cheered loudly.

Microdata co-venture

By then, Pick had primed the Microdata management with tales of his revolutionary database and what it could do for them. Microdata were excited but cautious. And it fitted their ambitious future plans. Until then, they had been a hardware provider, both directly to software companies, and via licensing of rights to build versions of their hardware. President Ham Hawkins had seen the future of software and rightly wanted a share of it.

Microdata's operating system was quite bare-bones—their software partners did most of the programming work. Now, here by chance, in their own backyard, was an OS complete with a database and the built-in ability to access it without programming. It might have been a match made in Heaven, and Pick turned on the power of his charm offensive to convince them of it.

After a few weeks of talks, Microdata agreed to up the ante by investing both sorely-needed equipment —two more computer systems and new monitors—and personnel to help Pick Systems. Three Microdata programmers joined the Pick ones, thus doubling our resources.

The stories from computer lore about 80-hour work weeks and all-nighters did not apply to us. Most of us were not “computer geeks” — single mindedly focused on the job at hand. We were more well-rounded than that, most with families or involvements, and many taking full advantage of the Southern California ethos: sun, sand and surf; hiking in the Cleveland national forest nearby; hang-gliding. It's a wonder that the group was as productive as it was.

Converting major operating system software to a reliable product is no easy task; and it was done by a half-dozen programmers in about a

year. This was high productivity by any standards. To be fair, much credit was due to Nelson's robust original design.

Paying Your Way

Within a few weeks of the Microdata influx, the original computer, now owned by Pick, was freed from its development duties, and began earning its keep. Timesharing the machine was a viable consideration, because the beauty of the Pick system in this mode was that all the requisite functionality had already been programmed into the base OS—it was inherently a multi-user, timeshared computer.

A local real estate company signed on to buy computer time on the Pick machine. The prospect of a newly-minted system, using untried techniques, a balky disc drive, and barely tested software actually earning revenue tickled Pick's fancy. And he welcomed both the income and the bragging rights of having a "production" machine before Microdata did.

Fortunately for Pick, the developer was both his friend and typically naïve on the subject of computers. Thus he was remarkably sanguine about the less-than-stellar reliability of his connectivity. Nowadays, when your Firefox browser "crashes," it gamely offers to restart that session, and, for the most part, will do so exactly at the point of the crash. My Mac PC can be forcibly shut down when I have a (very infrequent) hard freeze, and will restart every previously running program when rebooted. The Pick system exhibited no such resilience. When the developer's office called to say the connection had gone inactive, a couple of programmers had to be immediately dispatched to fix the problem, desperately trying not to lose too much information on the restart.

Soon after, the first two off-site production Pick systems were installed. One at a local clothing mail-order store, Contempo Casuals, and the other at the Bank of Guam. The latter attested to Pick's supreme confidence in the robustness of the software, as, at the time, there was no easy way for Pick Systems to offer software support to this remote location.

An ICE-y Tale

It was 5:30am. A knock on the door. Susan and I awoke groggily. We were living in Santa Monica, practically on its glorious beach, in a tiny one-bedroom that was half of a 1920's bungalow. Two men, grey-suited and red-tied, outside.

"We're looking for a "Chan...dra... Mertti?"

"Yes?"

"Immigration. We have reason to believe you are here working illegally on a student visa." Somehow I noticed they were not armed, or at least not obviously so.

"I'm not..."

"Sorry, we have to take you in to the LA detention center. Please get dressed. You are not allowed to take anything except your wallet and purely personal effects like medication."

I was, of course, still on my student visa of five years prior. But Pick & Associates had done the paperwork to sponsor me as what I thought of as being an "irreplaceable employee," including the dubious affidavit that I was the only individual uniquely suited to their job requirements. Ads had been placed in papers to support the notion that no American engineer was available in my stead. Did we answer them? Not sure. So I was confused, but, of course co-operative. The INS agents were quite friendly. I was driven to the LA detention center and placed a holding tank with what seemed like fifty others, all men, all Mexican.

I had my one phone call; luckily I had written down Pick's home phone and I woke him at 7:00am to tell him the news. He was furious. "What the fuck was the point of getting a lawyer if this shit happens?" He told me to hang tight and he 'd work on it. As he hung up, I felt profoundly alone.

We were processed into the system. Every single person ahead (and behind) me answered “Yes” to the question, “Will you accept voluntary deportation to your country of origin?” Except me. “No.”

A momentary confusion. “No? Hey, Charlie, where’s ‘em [incomprehensible] papers? Guy doesn’t wanna leave.”

I had a long conversation with one of the few others who spoke English. He told me this was the fifteenth time he’d been through the process—he’d come over, work for a while in the fertile California fields, go back, sometimes get caught. For him, it was routine. “We all do it...look at us!” as he translated our conversation and laughed with the others. He was amused by my angst, and tried to cheer me up.

“They don’t catch people like you every day! They’d have to keep you until your *jefe* got you out, yes? —they couldn’t ship you back to *India* tomorrow?”

I waited eight hours. I asked Charlie for something to read. He looked at me as if I’d asked for a double scoop of ice-cream. He did, however, bring me a two-day old LA Herald Examiner.

Pick showed up at 4:00pm, as I was biting my nails to the bone—by 5pm I was to be transferred to a center in San Diego, 150 miles away. He was remarkably calm though he had come some time before with cash bail only to be told Immigration did not accept cash or personal checks after 1pm. Cashier’s checks or Money Orders only. Banks then closed at 3pm, so he went to a local Post Office and told them he needed \$2,000 in Money Orders—maximum face value of \$50—so had to fill out forty forms.

“You’re my slave for life,” he informed me as we drove away.

Two years later, we had a hearing before an immigration judge, who castigated the INS for ignoring the fact that I, through Pick & Associates, had actually applied for a green card and was so immunized from further action from INS agents.

First Major Sale

In 1974, upstart Microdata was in competition with well-established Lockheed Systems to supply computer hardware and operating system software to the giant National Inventory Control Systems (NICS.) This multi-million dollar contract would've been a major boost for Microdata.

The Pick database/OS was considered vastly superior to the Microdata OS, so Microdata's senior executive Max Malone replaced the latter with the former in a last-minute decision.

Microdata scheduled a demo, involving high-level NICS personnel, and promised the ability to simultaneously query and update the database via seven monitors. Seven monitors, or terminals, on a \$18,000 machine in 1974 was considered beyond state-of-the art. Rather than have live operators keying in data, NICS accepted a computer-controlled, "robotic" activation of the terminals as proof of ability.

Microdata had recently switched to a new, "improved" disc system, faster (goes without saying,) and with increased capacity. The new disc "driver" software proved to have an unfortunate glitch which made it, in human terms, throw up its hands and go sulk in the corner. In other words, freeze—stop doing what it was supposed to do. This happened in every 10-15 minutes of hard, sustained activity, and was fiendishly hard to debug. The million-dollar-at-stake demo loomed.

At a review meeting there was a sense of extreme urgency. Fortunately for Microdata, Pick thought way outside the box.

He said, "The demo's for seven terminals, yeah? There's terminal zero doing fuck all." This was a spare monitor (called Zero,) normally quiescent as the "system console," only displaying esoteric system log messages and warnings.

"So?"

"Well, it could monitor the disk activity, right? It could check on the driver and see when it crashes..." as he stopped and gazed up at the ceiling, no doubt designing his program on the fly.

We others were still in the dark. A chorus of voices expressed their puzzlement. "What can it do?"

"Re-boot the fucking driver, that's what...hey, give me something that tells me the candy-ass driver has stopped, and I'll give it a kick-in-the-butt!" So this hidden program was to continuously check on the disc freeze. When found, it would activate a re-boot of the disc driver.

"But the terminals will stop display!" (As the disc activity froze.)

"They'll stop in a second or two anyway. This gizmo will start the disc back up in less than a second. Nobody will notice."

Another chorus of dissent.

"Fuck, these are marketing honchos. You think they are actually paying attention to the terminal display? Hell, we could probably show them random data, fucking baseball stats, any figures, and they'd be happy. " Pause. "Hmm, they'll probably have at least one techie with them, though, have to be a little careful."

"But Dick, when you re-boot, it'll sorta step back in time, right? Show the same data it showed a few seconds ago?"

"Yeah. So what? Acme Products, Acme Products, twice in a row, and they'll notice? No fucking way. Tell 'em the monitors are old and flicker a lot."

The kick-start program was put in place. At the demo, no one noticed. The duly impressed NICS bigwigs soon signed the contract. The disc driver glitch was found and fixed, as is the fate of all bugs.

Minicomputers in those days typically had a set of lights and switches on the front panel of the box. When events like the disc-driver freeze occurred, the lights would typically flash in a repetitive sequence, indicating an error. After the successful demo, Pick said that every Microdata/Pick system should be sold with a trained monkey whose job would be to watch for error sequences and flip the console switches

appropriately to correct the problem. Some of us programmers did not think this was a joke.

Computer Languages

A high-level computer language is a human-readable form used to write computer software. A specialized program usually called a compiler translates this to the internal form suitable for the destination computer. Source languages vary in ability, complexity and readability.

When writing this article, suppose I had to preface it with a short description of *every noun* I was going to use in the entire script:

Soviets: The country, and people, of the Union of Soviet Socialist Republics.

Satellite: A man-made object launched to circle the earth for scientific study.

etc.

I would consider this requirement onerous and impractical, and I would probably stop writing right there. But that's exactly what the widest used business computer language, COBOL, required you to do.

COBOL was developed in 1959, and a typical program had to have several hundred lines of verbose and tedious "description" before any actual instructions could be added. Amazingly, COBOL not only went on to become the most-widely-ever used language (it is estimated that in 1997, there were 200 *billion* lines of Cobol code,) but it continues to this day, albeit in a more modern form.

COBOL was developed by the US Army, and by one of the few women programmers of the time, Rear Admiral Grace Hopper ("Amazing Grace", and yes, she was in the Navy,) at the request of a consortium of six computer manufacturers and three government agencies. For years, Hopper had been interested in "English-like" computer instructions. Interestingly, Hopper had had to argue for years as to whether it was even possible for computers to understand English-like syntax, being

told at one point by her supervisor that computers only understand mathematics. Her view would have found common ground with Nelson's desire to talk to computers in a "natural language." Hopper and Nelson came up with radically different solutions to this requirement—she, a complex and time-consuming but powerful linear language; he, a free-form, English-like sentence-oriented communication. One wonders how their conversation on the subject would have developed if they had ever met.

The Pick database system in 1973 lacked a true language, other than the GIRLS pseudo-English inquiry. GIRLS had extensive and elegant change data functions, voluminously described in Nelson's wondrous flowcharts. But the Pick group simply did not have the resources to complete the partial implementation they inherited from TRW. Since there had to be a way for applications to affect data, Earl had quickly programmed a simple but limited way to do so. But we needed an actual language.

A Basic Language

Pick realized the need for a language simpler than COBOL, and one more oriented to business needs than the several other common science-and-math-oriented languages. He considered a simple computer language called BASIC (an acronym of the mouthful: Beginner's All-Purpose Symbolic Instruction Code,) originally developed at Dartmouth College and widely in use in academia.

Not everyone was a fan of BASIC. Edsger Dijkstra, one of the foremost computer scientists and a physicist by training, famously excoriated the language: "It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration."

I remember Dijkstra asserting in the preface of his seminal work, *A Discipline of Programming*, that none of his algorithms had actually been tested on a computer, a testament to the certainty of his theoretical approach. For me, and some of other Pick programmers who had no computer-science background, this was almost heresy. The computer code was tangible; its successful execution validated the problem.

Dijkstra's diametrically opposed view: "Programming is one of the most difficult branches of applied mathematics." It took a while for me to fully appreciate the connection of pure mathematics to programming, but I still cannot agree with that definition. And to be clear, practically all business programming, and even a large portion of systems programming, is a-mathematical.

The rational business world would ignore Dijkstra's rant: BASIC won widespread acceptance and was soon commonly distributed on business minicomputers, and, later, all micro-computers.

Pick also noted an opportunity in his new employee Simm's obsessive interest in Star Trek, which was written in BASIC. He challenged Simms to making BASIC run on the Pick system, claiming it would take months if not longer. Simms, ridiculing the time frame, rose to the occasion. He had recently studied a new methodology for writing a compiler (translator.) So he wrote a BASIC compiler in two weeks, and had BASIC running shortly thereafter. It was an notable instance of converting academic theory to reality.

Gus Giobbi, an early user of the Pick system, later the creator of the Pick conference world, had once characterized the Pick database as "the nirvana of the minicomputer world." He said of BASIC, "I can unequivocally say that Ken Simms saved everybody's ass who was trying to install [the Pick system]."

Simms had other priorities than mundane business applications. Star Trek ran slowly on Pick—it was a computationally intensive game, and low computation speed was the Pick system's major weakness. Nowadays, we can only marvel at the time and patience required to play this non-graphical version with the Enterprise battling eight enemy Klingon ships. "Battling" is used loosely, as, on each *typed-in move* by the player, the minimal display icons would disappear and the entire monitor screen would refresh after a second or two, in slow-flowing text. However, Simms could spend several hours a day pondering both his next move and how to speed up the Klingons' response, as he sipped his tenth diet soda.

The Rift

It's 11 pm. Do you know where your data is—or “data are” if you want to be pedantic? Nowadays, both the high reliability of computers and the tendency of files to be redundantly stored—like when you send an email with your child's photo attached, it's on the recipient's computer as well as your own—dulls the risk of loss. Cloud storage, or using Yahoo or Gmail, further encourages a laissez-faire attitude towards “backups.” In earlier times however, you had to backup your computer to off-line storage, typically tapes. It was not a matter of if, but when, the disc drive head, floating optimistically 1/100 of an inch above the rotating platter, encountered a dust mote larger than the gap, causing it to “crash,” hit the platter, and destroy itself. Power glitches and failures were also a major issue for those, like Pick Systems, without the wherewithal to install uninterruptable power supplies.

I came in one Friday morning a little earlier than usual. I walked through the empty offices to the warehouse area at the rear. There were our three computer systems, the hardworking timeshare, and the two development computers, TweedleDum and TweedleDee, humming and blinking purposefully. There were the ancillary tables of hardware-related scopes, wire-wrapped printed-circuit boards, tools, soda and beer cans, papers galore, power supplies. Posters of rock bands Led Zeppelin and the Grateful Dead were among many on the walls, reflecting the post-hippie music transition. A large rack to one side was devoted to magnetic tape storage (those 12” reels,) and one shelf in particular carried the fourteen “save” tapes which were updated daily, cycling every two weeks. Our entire software life was engraved on those tapes.

Except the fourteen tapes were not there. That shelf was empty. I stared for moments in shock, then calmed down; obviously someone (who?) had found a better spot to store the tapes.

I called Harry Emerson first: “What did you do with the save tapes? A little light reading at home?” He was not amused, and knew nothing about the missing tapes. Neither did our hardware tech. Seagraves, who always worked in the back, nor did the other programmers.

Time to panic. We started an immediate save on all systems, praying that we would not have a sudden power failure or earthquake — this was California, after all. We called Pick, who raced down to the office. He had realized what had happened: there had been ongoing friction between Pick Systems and Microdata, not least because Microdata's burgeoning sales of the Microdata/Pick system, now marketed as "Reality," meant increasing royalty payments from them to Pick. They had threatened to pull the plug on the joint development. They had just done so, in the worst possible way for us. Since the development computers belonged to Microdata, Pick knew they would soon come to collect TweedleDum and TweedleDee.

"Seal the doors," suggested Emerson, collecting a convenient piece of 2x4 lumber and hammer and nails.

"No need," said Pick as he apparently remembered something that gave him cause for joy, "God, what a fucking lucky stroke. I offered to buy those damn machines from them last month and they just sent me an invoice. Now where did I put that?" as he started rummaging around in his monumentally messy desk. "Here! I'll write 'em a check for the full amount."

Having done so, he turned to John Treankler, one of our younger programmers, saying, "John, walk this over to Accounts Receivable, will you? Look innocent, you do that sorta well. Make sure to get a receipt or something. I'm going to the bank, make sure they don't bounce it."

Sure enough, a clerk ignorant of the larger issues accepted the check and issued us a receipt for payment.

Not much later, Microdata sent an Orange County Sheriff, with a warrant, to demand the computers. Pick could gleefully wave the receipt to prove ownership. The puzzled sheriff retreated.

And we took Save tapes regularly out of the office for safe storage from then on.

The French Connection

Dodging this bullet may have been the best recipe for rapprochement between the companies. Pick and Microdata sued each other, and eventually reached an agreement: Microdata could market Reality under a much-reduced royalty schedule, and obtained a perpetual license for its continued development. Pick owned sole rights to the Pick Database outside that. Pick Systems was now established as a rival to Microdata. And Pick realized that he still needed hardware from them, so they were quite likely to put as many impediments as they legally could on delivery.

Microdata had licensed a French company, Intertechnique S.A., Plaisir, France (IN,) to manufacture and sell a carbon copy of their minicomputer. IN had a very successful business selling computer hardware even with the minimal Microdata OS to various French government agencies. They had heard of and were impressed by the Pick database.

IN's president, Edmond Marchegay, 50, had already initiated contact with Pick. He was an old-school French sophisticate, quiet in manner but purportedly steely in negotiation. Pick's charm offensive to woo Marchegay went all out. The rabid beer-drinker developed a sudden and quite informed interest in fine wines. When the French visited Pick Systems, there would be a French-vs.-California wine challenge. Marchegay had to reluctantly concede their essential equality after many bottles were drunk. On learning of Marchegay's other passion, sailing, Pick, an occasional sailor, rented a Cal 29 Bluejacket sailboat and took the French out through the pristine waters of Newport Beach into the Pacific ocean. Some of us programmers were reluctantly hired as crew, and gamely tried to comply with "haul that sheet up the jib, NOW!" and other incomprehensible commands. At least the fine wines were democratically available to all.

It worked—the two became good friends. Building on this, Pick soon had an agreement with Intertechnique to buy their hardware; in turn, IN installed the database on their systems.

Pick proposed an extension to the memory capacity of the IN minicomputer which would greatly increase its performance. The Microdata machine had a memory limit of 64 kilobytes. This hardware extension would increase that to one megabyte. Showing an unexpected capacity to design hardware, Pick designed it himself, and it was developed and soon manufactured in France. This was a major marketing plus for the Pick version compared to the Microdata one. The divorce of Pick Systems from Microdata Corporation was now total.

I visited Intertechnique several times to talk details about the software required to support this hardware extension. I then had the title of Vice President at Pick, though that meant little in the irregular way Pick Associates was run ("Board meetings? Yeah, we'll get to that soon!") The French, however, gave me royal treatment, including some of the finest meals I'd had. Made me wish that I actually had the power they imputed to me. I was also amused by the much more formal way Marchegay treated me in France, as opposed to his quite friendly demeanor in California.

Genius or Crazy?

Pick's one-on-one ability to charm people was not reflected in his general marketing savvy, which was a decidedly mixed bag. Even when, by the mid '80's, he had nominally relinquished control of Pick Systems (the successor to Pick and Associates,) by hiring one person after another to the office of president, Pick was firmly in charge.

He could be assured and persuasive in interviews. In one, he compares Peter Drucker's writings about an organization based on a hierarchy of people to: "Now, [when] we can have an information-based organization. There's nobody who's doing that yet. But that's the path we [Pick Systems] are going on ... It turns the Industrial Revolution upside-down, because [back then] you had to have a lot of money to build, say [a machine to replace a blacksmith.]... In an information-based system, it's easier to start a small organization ... it's the smaller units that are driving this sorta revolution."

Yet among his early infamous marketing images was his hanging upside down in “gravity boots” next to an IBM computer, labeled “Pick on IBM.” The image was widely distributed but it was doubtful if it actually furthered the fortunes of the database.

But by far his most outré marketing ploy was at the 1990 Comdex. Comdex was then the premier computer show, held yearly and populated by every major computer and computer peripheral manufacturer. By 1990, it had over 1,800 vendors in Las Vegas.

John Dvorak wrote in PC Mag: “Dick Pick, ... developed his company's Comdex exhibit this year featuring a rap theme song, break dancers and a portable radio giveaway.” The singer was none other than Pick’s fourth wife, Zion Pick, and the rap lyrics were collaboratively written by Jon Sisk and John Treankler, both technical types heavily involved with the Pick system, and Zion.

*Now I might have to sound a little disrespectful.
Just wanna school you on the Pick perspective.
A business solution that's really radical.
The people who use it border on fanatical.
We fondle your data, it isn't so hard.
Not looking at the world through holes in a punch card.
Pick's got the power to break the chains.
Rearrange your brains, eliminate the pains.*

Whether this was genius or a crazy stunt, the performance stood out even in the carnival-like atmosphere of Comdex Las Vegas. Pick claimed he got over 4,000 leads from there, a figure not verifiable but certainly the Pick booth was jammed with attendees to the very end of the show.

Power to the Pickies

The power of the Pick system was rooted in its ease of application development, and the extreme efficiency of its internals, which allowed an economy of use far surpassing rival systems. This was particularly

important in the days when programming minicomputers was relatively expensive.

The Pick database appealed to thousands of small businesses, owned by individuals who were willing to take a chance on software that bucked the trend. There were, of course, many large companies that bought into Pick—ADP dealer services, for instance, a company estimated to write one-third of all payroll checks in the US.

Many smallish companies using Pick—10-100 employees—would not even have a resident computer expert on staff, a marked contrast to almost every other computer system of the time. Typically, someone on staff would learn the database access language with a short week's training at Pick, taught usually by wunderkind showman Jon Sisk. They would then be able to extract data and produce customized reports. The more adventurous would start writing programmed instructions in the Pick script language (cumbersome but easy to learn,) and, eventually, in BASIC.

The ability to “port” the Pick system became an important feature in the ongoing success of the database. Since it was written in its own language, it was not tied to any particular computer hardware. For a while, other firmware versions were developed, notable for the Honeywell Bull Level 6, the first mainstream computer that ran Pick. Soon, minicomputers became fast and flexible enough that a mere conversion (almost like simulation,) of Pick instructions to whatever hardware was being used was enough, greatly speeding up the transfer to new hardware. The parable of the fast small car and the slow large van no longer applied—the car's immensely increased speed made it outstrip the van's productivity.

The Pick database went on to increasing success. Microdata continued to sell Reality in increasing numbers, and was eventually bought by a British company, Northgate Information Solutions. NIS now claims 15,000 installations with 200 of them being Global Fortune 500 companies.

Intertechnique enjoyed enormous success with the Pick database for the next three decades. As late as 2005, the French publication *Agora Vox* maintained that a majority of French libraries and auto dealers, and the ministries of justice and agriculture, used the Pick database.

One of the Pick “dealerships”, akin to a franchise, the grandiosely named The Ultimate Corporation, reached sales of over \$200 million with a total staff of only about 170 within three years of its incorporation.

Pick went on to create a world-wide dealership system. Various large computer manufacturers licenses such as Fujitsu Systems and Siemens bought Pick licenses.

By 1992, the Pick marketplace worldwide was estimated to be 3 billion dollars, with over 3.5 million “users” and over 4,000 applications. The largest Pick installation supported over 2,000 users on an IBM RT, not a minicomputer, but nowhere near the size and cost of a mainframe either.

Within ten years, those numbers had halved.

Post Mortem

Why did the Pick database not achieve the lasting success it deserved?

Most importantly, by the late ‘70’s, other competing databases had developed. All of them followed the concepts introduced by IBM fellow E. F. Codd, which he called *relational databases*, underpinned by a branch of math called relational calculus. And this was a slow but crippling blow to the Pick database, developed as it was by non-mathematicians and having no academic “white papers” or leading authorities to bruit its authenticity. The Codd formula naturally found strong support in academia. As a result, within a decade, the typical computer science student would have been steeped in the Codd “relational” orthodoxy, and the Pick database would seem quaint and primitive by contrast.

In an incredible coincidence, Pick met Don Nelson on a beach in Hawaii in 1982. Nelson had, once again, quit a well paid job and drifted. Broke and almost homeless at the time, Pick offered him succor: a job at Pick Systems with only the vaguest charters to continue development of the underpinnings of the database. An unspoken need was to come up with a more theoretical description of the database, to counter the competing databases' argument that Pick was inauthentic and unreliable. Unfortunately, Nelson never lived up to his previous potential. He produced a few papers arguing for difficult and not-quite-needed extensions to the database structure, but little else. Nevertheless, Pick kept him on the payroll for well over a decade.

Other theories on the Pick decline abound, but it might make a case-study in how not to market a product: A business marketing via license should have a strong presence and reliably even terms. Pick had neither. He allowed his licensees to do what they willed with the product, and did not even require the name "Pick". Every one of his licensees had their own name—the names Universe, Ultimate and Climax memorably stand out—diluting the market presence of brand Pick. And while everyone acknowledged the ability of Pick and his programmers to move the product ahead, he neither enforced standards nor required his licensees to incorporate his new features.

Gus Giobbi had created the Spectrum Conference and shepherded its success through the '90's. He also started an association called Spectrum Manufacturer's Association in the late 80's which attempted to recruit the various factions under a common, "Multi-Value" name. It seems supremely ironic that even an organization whose avowed purpose was to uphold the Pick brand did not have "Pick" in its name! While most licensees joined, it was mostly lip-service; neither a common standard nor the means of enforcing it emerged.

Finally, when Graphical User Interfaces (GUI) appeared, Pick strangely clove to a extremely retrograde idea: that the WIMP interface, as he no doubt was happy to call it (a short-lived acronym for Windows-Icon-Menu-Pointer,) was never going to catch on in business computing because you could enter data faster with a character (typing only) interface than with WIMP. The Pick programmers tried long and often to

change his mind, but failed. The Pick database from Pick Systems itself never developed a graphical user interface, though all the licensees created or adopted GUIs.

Coda: Full Circle

The company Pick Systems was bought out and changed names several times. Its latest incarnation is Rocket Software, which owns many of the non-defunct versions of the Pick database and sports a very impressive list of top-level clients. It is valiantly trying to merge the different versions into one. Taking a cue from revisionist history attempts the world over, the word “Pick” does not appear on Rocket’s website.

At the time of the Codd revolution, there were non-relational databases other than Pick. Four decades later, another seismic shift may be occurring. Many non-relational databases now are listed under the banner “NoSQL” (SQL is the acronym for the main data access mechanism for relational databases.) NoSQL databases violate many of the key precepts of Codd’s relational databases—shades of Pick! There are, by now, over a hundred NoSQL databases. Facebook, Google and Amazon all have embraced NoSQL databases, so the sheer weight of these companies will ensure the ongoing development and growth of NoSQL.

The Pick database variants have made it back into most NoSQL webpages, documents, and wikis under the category “Multi-value”.

Pick is dead; long live Pick!

The Players Now

Richard A. Pick died of complications after a stroke incurred while tending to his beloved kelp farm off the waters of Newport Beach, on the 19th October, 1994, aged 56. He is survived by three of his wives, and three children.

Donald B. Nelson died in 1998.

Philip Earl retired after a long second career in photography and lives in the same house he bought in 1970 with his wife of 51 years. In '03, he directed the acclaimed short film, *The Kids' Guide to the Internet*.

Ken Simms died in 1983 after a long battle with colon cancer.

Harry Emerson is retired after being a long-time consultant on the Reality database to various Pick licensees. He now spends much time in doing research on the Russian dance scene, bi-coastally following Russian ballet.

John Treankler still works as a consultant to Pick-owning systems. He lives in the Midwest where he cannot see his neighbors, a far cry from his previous Orange Country suburban home.

Jon Sisk is semi-retired and lives in in Virginia, having tired of teaching the great unwashed masses the beauty of the Pick database.

Gus Giobbi moved out of the Pick world, relinquishing control of the Spectrum Pick conference to others. The Spectrum Manufacturers Association is defunct.

Zion Pick inherited control of Pick Systems, then sold it shortly afterwards, and is a real estate dealer in Orange County.

Chandru Murthi is mostly-retired in Brooklyn, NY, having never left the Pick "world," thus having the dubious distinction of working in the same field for 48 years. He has dabbled as a Theatre Director, and, armed with a late Master's degree in Urban Systems from Pratt Institute, is an occasional Sustainability Consultant. His son, Dylan, 20, exhibits not the slightest interest in either computers or engineering.