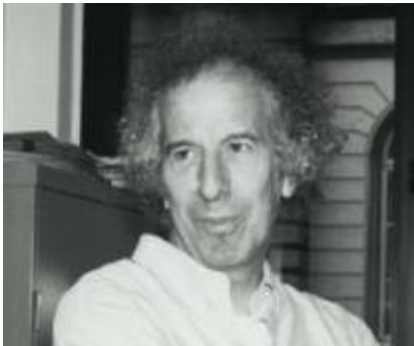


Map of/Excerpts from Oct. 30, 2017, Conversation between Yohanan Spruch, Oren Beit-Arie, and Jerry Specht

[Link to complete unedited \(1 ½ hour\) audio recording of conversation](#)

(Firefox and IE work better for me for moving around the recording; the Chrome bar for this is too small.)



Yohanan Spruch



Oren Beit-Arie



Jerry Specht

Interview map/excerpts

h:mm:ss

0:00:45 Yohanan's background: graduated from Hebrew University of Jerusalem, worked with Hebrew University IT department)

0:01:20 Graduated (in 1970), started to develop the IT dept with the Ministry of Education, and took a computer course.... And then, as a member of the IT dept, there was a need for a library system.

0:02:30 Jerry: "Oren and Ron Biron said that with the IT Department you had developed the Student Administration System and that there was a database that you created as part of that which you took and used as part of Aleph."

0:02:50 Yohanan: "The Hebrew University had a Control Data computer.... For student administration we created database system written in FORTRAN"

0:04:00 Yohanan: "... There's a story... The old software was developed in a coffeehouse ... at the old campus of the university. And then after having this database we used the database to develop the first version of Aleph on the Control Data machine... The first Aleph was sold on Control Data machine to CSIC Spain [the Spanish national research council]. And together, with the software and hardware, they got the library system. [in 1982] ... The restaurant for CSIC was

in a nice house where there was a piano which Lorca [the Spanish poet] played when he was a student (in the 1920's)... Those are the things I remember.”

0:06:05 Yohanan: “And then Control Data tried to sell the system to ETH Zurich,... but they didn't like the database, it wasn't good enough according to their standards, so they didn't buy Aleph then.

0:06:40 Aleph running on shared CDC machine at Hebrew University, but, in Spain, CSIC CDC machine was a dedicated machine(?)

0:07:20 Jerry: “The system I worked with before Aleph was NOTIS (Northwestern Online Total Integrated System)... It was kind of dying out at the time that Aleph was coming to North America.”

Yohanan: “To come to America to replace NOTIS; that was our main idea.”

0:08:00 Jerry: “The development of that system began in 1969. The developers looked at COBOL, but dismissed it because, at that time, it lacked support for lowercase characters, diacritics, etc. They ended up going with IBM 360/370 Assembler, which gave them the flexibility to do mixed case, diacritics, etc. – and to achieve very good performance --, but tied the system to the IBM mainframe which was OK during the 1970's and '80's, but became a real problem with the advent of client-server in the 1990's. Were the Aleph programs written in COBOL? You mentioned that other language [FORTRAN] or were the Aleph programs written in COBOL from the first?”

0:08:50 Yohanan: The Control Data machine was 6-bit characters, so in order have Hebrew and English you had to develop something similar to Unicode. Because the 6-bit character set was *very* limited. So we developed something which would use 12-bit. That was enough.

0:09:50 Jerry: “As far as the language goes,... did you do that in COBOL or something else?”

Yohanan: “In CDC the main program was COBOL.

Jerry: “At some point, I think it was quite early, that you also started writing programs in “C”.... Do you remember just when that started?”

Yohanan: “No. We started “C” from day one because we needed ... the operating system, the ability to share multi terminals ... we didn't use any terminal software ... In IBM there was a lot of terminals connected together. It was controlled by a system called ... CICS. So we prepared something similar to CICS, together with Control Data.... That's one thing we developed. The other problem, in order to be able to display the character set, we developed our own software on an Elbit terminal. Oren, do you remember?”

0:12:30 Yohanan: “CICS was working in what you call the block mode.... Send the whole screen at once (rather than character-by-character). They developed a special chip for us that would work with Hebrew, English, and Arabic on the same chip. It was the old, old, old days.”

0:13:30 [Northwestern (NOTIS) having to do the same things with terminal handling that Yohanan describes prior to IBM's introduction of CICS (in 1975).]

Yohanan: “The same problem exactly.”

0:14:00 Jerry: “Were you able to use most of the existing CDC Cobol code when you moved from CDC to VAX/VMS – and I suppose the “C” programs too – or did a lot of it need to be rewritten?”

0:14:20 Yohanan: “A lot was rewritten, but we were able to use a lot of code too.... With the structure, a lot of it was the same. You know Aleph has a lot of tables ... and all the tables were the same.... The concepts which are part of the system were unchanged.... I don’t remember the quantity, but, later, moving from VAX/VMS to Unix, it was the same code....

“With VAX/VMS it was VT100 dumb terminals. Nothing was done on the terminal;... everything was done on the server.

“But still we used our own database, indexed-sequential system. We didn’t use any database system.... We built the I/O layer in COBOL and “C” from the first. All the connection between the Aleph system and the computer system was done in “C”. Amos Aner wrote all the software which worked on the Control Data. He wrote together also the “CICS” on the Control Data. The “CICS” system for Control Data was written together with us and Control Data....”

0:19:30 Yohanan: “Then we decided to move to the VAX machine.... We were in competition with the university in Haifa.... They started to develop their own system on a [DEC] PDP machine.... We competed over which would control Israeli libraries. In the end, they decided that Aleph would be the system adopted by MALAG (the Israeli Council for Higher Education). The Israeli libraries paid half of the regular price.”

0:21:25 [Yohanan’s indexed-sequential] database system, used with Aleph on the Control Data machine, was brought over to VAX. The student administration system was also moved from CDC to VAX/VMS. Yohanan wrote an simulator.

0:24:15 Yohanan: “When we moved to the [VAX] VMS, because we used the I/O layer, we could change easily....”

Oren: “If I may say so, one of the brilliant element in the design, the architecture of Aleph ... was that I/O layer that basically separated the machine architecture from the software. That transition was made possible,... and later, when we transitioned from VMS to Unix, and later, the move to Oracle.... It was much easier.”

0:25:05 Yohanan: “We worked a lot with the I/O layer. The programs didn’t communicate directly ... but with this well-defined language....”

0:25:30 Oren: “We didn’t use fancy language then but, today, we’d call it APIs.”

Yohanan: “A lot of things [later] become standard-industry ... with a different name.... X-Services.... We tried to develop an open system; X-Services was an open-system approach....”

Oren: “I think that the way you organized the system, both internally and externally, had this notion of APIs: ... internally, this I/O layer separated the system from the I/O; and externally there was the x-server, which was kind of an API....”

0:27:05 Jerry: “Within the Aleph programs you had the edit_field/edit_paragraph/edit_doc tables and tab_expand/tab_fix, kind of like building blocks. Not every customer would use those but especially some of the more sophisticated customers were able to take them and make their

own structures. There's a lot of flexibility there. Much more than other systems.... Those were things that you had the idea of putting into the system, Yohanan, am I right?..."

0:28:15 Yohanan: "... I don't like customization.... I tried to design a system so the customer can do what he wants without bothering us. That was the idea. Everything is table-driven. From Day 1 we decided that Aleph should be an interpreter and not a ... hard software. All the controls were in external tables which the end-user can use to tailor an environment which he likes."

0:29:00 Oren: "Yohanan, this is something I remember really strongly as a junior developer: you really hated special cases.... so we don't have to rewrite code; it was all about generalizing.... Some would argue that maybe we took it a bit too far, and at some point it was so generalized it was hard to manage... It's really extraordinary how flexible the system was."

0:30:40 Oren: "The clever thing about this when I think about this, Yohanan -- it's a huge system that was written by basically one person.... But no one person, not even you, can write all this code.... Yohanan, I think that what you created is basically a *tool* for others to come in and do all the hard work: fine-tuning, setting up all the codes, the circulation policies, subfield values.... A tool, using their time, their effort, their resources to make that tool work to do specific tasks in libraries.... A way to scale up the work by giving them means to actually make it work for them without them having to bother you with coding.

0:31:40 Yohanan: "The basic idea was ... same code for everybody."

0:32:10 [VMS to Unix; early '90's.] Yohanan: "Then we had the same code for two systems [VMS and Unix]. There were shared lines and lines with "U" and "V".... The same code was compiled on the two different systems.... It lasted for a long time ... on both systems.

0:33:15 p_file_08 (file-08 / fix_doc_do_file_08) Generic fix routines.... Another "toolbox". Quite powerful.

0:34:15 Yohanan: "The idea was ... a table not of data, but a table of routines.... We started using Micro Focus (COBOL). It was a real-time linker. Normally you compile everything and you link it in advance; then you run it. Micro Focus COBOL could do it on the fly. The table was the names of routines which was loaded in real time,... activate this routine instead of another routine:... filing routines, check routines, and so on.... You don't have to ship the whole system.... You can send the routine with the name and that's it.

0:36:20 Oren: That was critical. This is a huge system.... If every time you had a fix routine you had to compile the whole system -- you know in the '90's -- it would take you hours.... The other thing I wanted to quickly say about this architecture that Yohanan designed across the system, that was what made us a contender even in the U.S. Our system wasn't ready for the U.S. market. Let's face it; let's admit it. We didn't even have the proper MARC structure. But it had such a great flexibility.... Yohanan, I don't know if you remember, but in the early conversations with the U.S. customers like Harvard and Notre Dame, you always claimed, "Of course we have MARC. We have everything... You can write whatever you want there." For them the big fact was that they didn't care about the generalization; they cared about their usage... The way it was sourced was brilliant: because the system didn't change its architecture.

It was built in the architecture ... and just start building all those check routines, fix routines ... they were integrated into the system but without having Yohanan core development for each one of them. It was just integrated links in real time and executing.... That was the real essential component in getting into the U.S. market I think....

0:38:30 Jerry: With an integrated library system there are different moving pieces, a lot of parts that need to fit together. And different functions are challenging in different ways, but I think that a good Keyword/Boolean function is one of the main challenges which designers/programmers of systems for large research libraries face. I feel that the Aleph keyword/Boolean is very well-constructed and performs quite well. It seems to me that the z97/z98/z95 and the z980 are very well thought out. And that the z980 was introduced to allow something approaching real-time updating -- since the direct update of the z98 would take too long.... Is that how that happened?

0:39:40 Yohanan: The goal was performance. The problem was how to use the ... whole 24 hours of the computer, not just the 8 hours of the day. The idea was to divide the operations into high priority and low priority, and try to do the low-priority [at lower-use times]

0:40:30 [Jerry] NOTIS / BRS STAIRS [keyword] software. "There wasn't any ue_01. They would have to periodically rerun the whole indexing."

0:41:15 Yohanan: "... If you want to update an index, with the multi-user, it's impossible. One user should be able to update the index. Otherwise, you end up with a lot of locks, and it's impossible to work.... And the time they are running [making use of the night]. Those are the two things. The whole idea is that today normally you have a server and you send a message to the server; do it. The same server does it for *all* of the users.... Every user when he goes to the database locks the database into an update.... ue_01 solves this -- the single process manages everything.

0:42:30 [Jerry] Difficult problem of proximity searching; later, use of word-pairs/double-words for adjacency; but z95 bitmap elegant solution for proximity

0:43:48 <pause for break>

0:44:00 Oren: "... a number of core components that were imagined, really, by Yohanan, way before they became standard in the industry....

Parallel processing was invented in Aleph.... This was brilliant! Obviously we didn't give it the right name, so ..."

0:44:45 "We mentioned earlier the APIs, I/O layer, X-Server, the open platform, but parallel processing is another great example how in COBOL, in "C", using ... very basic tools ... for queuing, breaking up processes into sections ... really brought in tremendous power. It's interesting to think how much of that was actually invented in real-time, while presenting to potential customers. A lot of that was in the context of BNF (Bibliothèque nationale de France) ... I think it was one of the most important "failures" [-- failure to get them as a customer]. And later when we started communicating with people like Dale Flecker at Harvard, and Notre Dame,... when they presented with problems, on the fly, stuff was invented. What's absolutely

amazing is that here we are, 20/25 years later, and there are still thousands of customers around the world using that software, day in and day out....

0:46:40 Jerry: "It's interesting you should mention Dale Flecker because ... he, along with Nolan Pope [at Wisconsin], are two of the main reasons I ended up working with Ex Libris.... They agreed that Aleph was something that could work in North America...."

0:47:50 Oren: "... A lot of what we learned to get Aleph to do what it needed to do in this country was the result of conversations with people like them.

0:48:10 Yohanan: "A customer who knows what they want really helps us to develop the system...."

0:49:00 Jerry: "It's really important to have good [smart] customers.... If you have customers who make unreasonable or completely unique demands, and you spend a lot of time developing it, that can be a problem,... But if you have customers who understand the needs of large academic libraries and can direct us in developing them, that's critical.

0:50:00 Yohanan: "There was the idea that you have to go to the big ones, and then you go down, not the other way around.... which helped us."

0:50:30 Oren: "They saw the intention being executed. We stood there and talked about something and waved our hands,... Sometimes as early as the following morning we could demo stuff.... The quick turnaround.... Of course we had a very methodical release control, but the ability to code quickly and show results was also critical.... The ability to execute."

0:51:20 Yohanan: "Today Omri is working with what you call "agile" system.... Our system was agile from day one.... The software was written in such a way that it was easy to change, easy for newcomers to go in and do what they had to do ... and easy to add more features into the system...."

0:52:40 Jerry: "And the performance too. Not just the flexibility. The system had very good performance. That's not a given. Sometimes people can have a lot of functions in the system but they're just way too slow.... Almost across the board,... Aleph has had really good performance.... I think you've had a lot to do with that.

0:53:40 Yohanan got two programmers just out of high school who wrote the programs in "C" language.... And today both of them are professors in the U.S. One is Physics [Yossi Farjoun]; one is mathematics."

Jerry: "... You picked some good people."

0:55:40 Jerry: "As far as performance goes, with the Keyword/Boolean, for instance, the University of Binghamton, part of the SUNY system, was able to successfully regenerate the Aleph Word indexes on their database of 3.2 million bib records, using p_manage_01, this was several months ago, in five hours. You didn't used to be able to do that, but the improvements in the I/O performance, the hardware, the disks has helped with that, but ... that's using the programs as they were written originally...."

Yohanan: "Twenty years old programs ... but still running."

0:56:45 Jerry: Another complex, important function in the system has to do with authority records, cross-references; complex interactions between the authority doc records and the

bibliographic headings ...: ue_11, ue_08, ue_01. How would you compare that to Keyword/Boolean? Which was harder?..."

0:57:35 Yohanan: "... Oren, part of the authority control was developed together with McGill. Right? You remember?... McGill had a lot of demands for authority control.... We went together, Oren and me, to McGill... And Naomi ... or Judy?... We didn't understand anything about authorities then... At McGill they were at a multi level of authority control: French, English, [local] ... And then we developed the multi-lingual authority control. It was a nice time...."

0:58:30 Oren: "... We worked also in the EC,... and I think it was a few years earlier."

Yohanan: "We *completed* with McGill."

Oren: "We had already the basics.... When we came to [North America] we had the basis for a lot of stuff. But it was more of a tool than full-functioning.... They had good systems that we replaced, mature systems - systems that were able to do a lot. I don't think that was the case to a large extent in Europe. Same thing with much better, clearer specifications of their needs as well. That gave us a tremendous boost. And the ability to take the infrastructure that was created beforehand and apply it."

0:59:40: [Jerry] Authorities more important for North American customers(?)

1:00:20: Yohanan: "When you go to developed markets and you know how to do things that leverage on your software ... and to develop a better system."

1:00:50 Oren: "... Keyword search wasn't something new.... I think what was really unique to this system [authority control] and had to be written from scratch.... In keyword, maybe the proximity search [was entirely new]... And then the authority control and **browse**. Browse was a feature that wasn't existent in other systems,... so that had to be developed from scratch."

1:01:55 Yohanan: "... I hope that vendors today when they develop systems they write more documentation about features than we have in the past. In twenty years, when you talk about Alma, there'll be more information."

1:02:20 Oren: "... Do you know how the first documentation of *Aleph* was done?"

Yohanan: "According to the contract between the Hebrew University and Ex Libris, we had to supply the documentation by ... some date. There wasn't any documentation, so we can continue from there...."

Oren: "When I started, there was no documentation. Yohanan basically told me: get in the program and read it. It was in COBOL."

Jerry: "It *is* readable."

Oren: "But beyond the program there was no documentation ... about data model.... What Yohanan did, he rented a beach house in one of his favorite locations in Israel and he went there with his wife Daphna, for a couple of weeks or three weeks, I think.... This was before Internet, so he took a VAX/VMS and a terminal.... I imagine you just sat in front of the terminal and went program by program and just talked and Daphna wrote it down.... And we're talking about the mid-nineties. Hand-write...."

1:05:00 Jerry: “With the NOTIS system, which was written in IBM Assembler ... more cryptic ...” extensive comments were needed – whereas COBOL is more self-explanatory.... “I had never even *looked* at a COBOL program at the time I started working with Ex Libris and Aleph.... But over time I’ve developed a pretty good understanding of COBOL and the Aleph programs that are written in COBOL....”

1:06:10 Jerry: “As you may know I interviewed Barbara Radel. She said during her interview: ‘Aleph was developed on CDC and then it was ported to VAX/VMS. Then everyone started talking about Unix. There was a very successful company in the UK. I think it was called ‘Libis’. They were on VAX/VMS. When we were trying to break into the UK, they had cleaned up the market. There was the same sort of thing in Denmark. Ex Libris was the only library company that managed to port their software from VAX/VMS to Unix. This is where you can see the brilliance of Yohanan. The UK company fell by the wayside and the Danish company contacted us and they became our distributor in Denmark.’ [end Barbara] I think there’s some truth to that.” In the case of NOTIS, written in Assembler, certain concepts were carried over but the code had to all be rewritten – with Voyager being the real successor to NOTIS.... “But it can be a big job: to take a well-developed, large [integrated] library system and to convert it into an entirely new language, environment. So I think the conversion [of Aleph] to client/server was a big and difficult thing, but I think it was done, all in all, very successfully.”

1:09:10 Yohanan: “Again,... the whole system was based on an API approach ... relatively easy to add client-server to dumb terminal approach.... There were a lot of nails on the server [you could hang things from] ... API approach ... we used it.”

1:09:50 Jerry: “I’ve met you in person several times. I remember meeting with you in Jerusalem, I believe in 1999.... I remember that one of the main topics of discussion was the HOLDings record – and its relationship to the item record. The HOL record existed prior to Notre Dame and North America, but wasn’t connected to the items in the way we felt was needed: an automatic connection so that the update of the HOL 852 field would automatically update the items associated with that particular HOL record. Do you remember our discussions of that? I think it was very important.... It wasn’t something that previously there had seemed there was a need for.... What do you remember about that?”

1:11:10 Yohanan: “I don’t remember it, but I think the HOL was something new from the States.... There was the bib record and the item record, but there was some information that was in-between ... and you pushed us to develop this.... But the idea was the ability to move fluently between... You don’t want to update the thing twice.... You helped us to put these things together, to develop a better system ...”

1:12:15: [Jerry] Later, with Harvard ... idea of automatically generated holdings summary from the items themselves.

1:13:05: Yohanan: “... both directions: from the holdings information into the item and from the item into the holdings.”

1:13:30 [Jerry:] Convenience of having update to holdings record be propagated to associated items.

1:13:50 Yohanan: “... It’s the same data in a different format....”

Jerry: "I think that, all-in-all, what we've come up with has worked quite well."

1:14:30 [Wrapping things up....]

Jerry: "Thinking back over your years with Aleph and Ex Libris, what really good decisions did you make? What did you do really well?"

Yohanan: "The first good decision was to pick Oren for the company...."

Jerry: "I agree!"

Yohanan: "The second good decision, later, I met my wife Charlotte at Ex Libris, so I married her. It was a good decision...."

"... Building the Aleph system ... the democratic spirit ... because everybody had his right to object and to change, or to argue.... The spirit of Ex Libris allowed us to build Aleph, I think.... It was a different company. That's the best thing I've done at Ex Libris, I think."

Jerry: "... That kind of ability to recognize good ideas and be open to them is very important...."

1:17:15 Yohanan: "They've been good years.... No, Oren?"

Oren: "Yes, good years.... You know, Yohanan is too modest.... There was Yohanan's relentless innovation, but also ... *passion*. You know Yohanan's passion was sometimes challenging in a lot of battles, but I think that was contagious.... And also *customers*. I think customers who met Yohanan and kind of sitting in one room,... they came out saying "Maybe I didn't understand half of what he said, but I want to work with this guy".... And the *result*.... As I said before it's not just talking, it wasn't just waving hands.... As early as the following morning they saw pieces of code.... Yohanan managed to get from all of us 150%. You know, so everybody worked hard and everybody kind of felt the challenge and wanted to excel, but also ... Yohanan always had the balance of working hard ... but having fun.... Rewarding hard work with, you know, going to a good restaurant, drinking good wine. So it didn't feel as if it was, you know, for nothing... Plus,... it's amazing how in COBOL, which is almost the most ancient language, we were managing to do such optimizations. So it felt good! It was ahead of the curve. "

Jerry: "As I've said before, programming language is greatly overrated as a factor in the quality of the system.... I think we covered a lot, so that's all that we really need..."

1:20:15 [*The End*]