Oral History of Burton Grad (IBM)

Interviewed by:
Ed LaHay

Recorded: November 28, 2007
Mountain View, California

CHM Reference number: X4362.2008

Table of Contents

Burton Grad (IBM)

Conducted by Software Industry Special Interest Group—Oral History Project

Abstract: Burt Grad (in one of four interviews) discusses his experiences in IBM from his joining the company in 1960 after leaving GE. He covers a wide range of assignments during that period including his initial work on the Study Organization Plan and Decision Tables. He discusses work on operations research applications like linear programming, critical path management and simulation. He talks about his work with the IBM Science Centers and the use of VM/CMS as a principal timesharing system along with the release of APL and HASP. He reviews his work on the IBM Unbundling Task Force and the announcement of the first priced programs by IBM. He then became a Development Director in the Data Processing Division and discusses some of the principal products like CICS and IIS as well as IBM's failure to make any successful application products for large computer customers. Then, he covers the international software development work at IBM and his reassignment to IBM Research and the technology transfer projects that he worked on there, albeit unsuccessfully. The interview closes with his decision to leave IBM and start his own consulting practice in 1978.

Ed LaHay: This oral history interview is being conducted by Ed LaHay, and it's an interview with Burton Grad, taking place on November 28th, 2007, at the Computer History Museum in Mountain View, California. Burt, thank you for sharing your IBM experiences with me. Would you talk a bit about what it was that got you to leave GE and come to IBM, how you got to IBM, and who the first group of people were that you worked with.

Joining IBM

Burton Grad: Sure. I left GE primarily because GE wasn't going to go ahead and implement some of the things I'd been working on, and that was somewhat frustrating. So, I was looking for a change, and IBM was the obvious company to go to. I didn't apply to anybody else. I looked only at IBM, and interviewed at two different places in IBM. I'd had contacts with IBM sales people and they introduced me to two people at the company. One group was in the Data Processing Division under Bob Bemer. And Bob had a small group in White Plains, N.Y. that was doing advanced studies in the use of computers, computer languages and so forth.

The other group was a more advanced group up in ASDD in Mohansic, N.Y., under Vilar Kelly. And they were doing advanced studies on how computers might be used. That was a tough decision for me. Both groups made me an offer, and for whatever reason I decided to go with Bob Bemer in DPD, the Data Processing Division. Bob had an incredible group of people working with him, and I was certainly proud to be part of something like that. One comment I've made is that, even though GE was an absolutely top notch company and had great people to work with, I didn't find people of the same quality level at GE as I did at IBM. In this one little group of a dozen people who worked for Bob Bemer, seven or eight were just world class in their technical abilities and how smart they were and what they could do. So that was the first group I went to work for at IBM.

LaHay: Did you feel challenged when you joined that group?

Grad: Oh, yes. I came in, of course, as an experienced hire, where normally IBM employees came in at the ground floor, sales or systems engineering, but this was a professional group that I came into. I had felt very successful at GE, so I had a lot of self-confidence. But I was challenged pretty early in terms of what have you got new to offer us? They had me working right away on a commercial language translator called COMTRAN which was supposed to be the commercial equivalent of FORTRAN. And I hadn't done much work on language development before, so that was a challenge, what the commands should be, how they should be structured. That was a new thing for me. I started working on that as one of my first projects.

LaHay: So did COMTRAN ever see the light of day?

Grad: COMTRAN never quite made it. COBOL took over. I was involved actually as one of the people from IBM who helped to work with the Codasyl committee on the COBOL specs. We were trying primarily to make sure that it had the COMTRAN capabilities in it. We felt at that time that we were unhappy that COBOL was not as good as COMTRAN since we felt that the structure of COMTRAN was cleaner. COBOL was designed by a committee and it reflected it. Conversely, it did become the standard for the next 50 years, so somebody knew what they were doing.

LaHay: Apparently. So what other key projects did you work on when you were in Bemer's organization?

Study Organization Plan

Grad: At GE I had worked on two major projects. One had been on decision tables, the ability to express decision logic in tabular form instead of in a flow chart or in a decision tree.

And the other had been how to design large scale automated systems, both office systems as well as factory systems. With decision tables, I had been told by GE not to publish that material. That was one of the reasons that I was unhappy there. They wouldn't implement the use of decision tables and they wouldn't let me publish what I had done, and the same thing happened with the automated factory work. However, I still felt that there was a way of going about the design of systems that didn't have to be factory automation; it could be just the office side of the thing. And so I proposed a project to Bemer's boss at the time, John Harmon. I told him that I had this idea of how one could organize a study in a business so that you would have a total application, that it wouldn't just be part engineering or part accounting or part manufacturing, but that it would look at the entire business paperwork, the whole information processing cycle. He thought that was too ambitious an idea and that I probably couldn't do it. Well, I had done my homework. I had talked to people in some of the sales branches who had some major customers that they were trying to work with, and so I said, "Well, why don't you call some of your branch people and see what they say?" And of course they were positive since I had prepped them. So, Harmon said, "Fine." I had only been with IBM for about two to three months and I was told, "Okay, you're a manager now, and you're going to have this team of people and who do you want?" And I took 3 people who were working on Bob Bemer's staff, with his agreement, and they became part of the team: Tom Glans, David Holstein and Lee Baker. So I had three people and a secretary working for me within a few months of joining IBM.

LaHay: So was this what led to the SOP [Study Organization Plan]?

Grad: This led to what we ended up calling the Study Organization Plan. We started with an idea that was left over from what I had done at GE, the idea that you could look at the business as a whole, by looking at all the different pieces from engineering through manufacturing engineering, through cost accounting, through accounting, through billing. There was a flow that one could look at as an entity, and instead of trying to deal with each individual part we could optimize the total flow; rather than come up with a solution for engineering or a solution for manufacture engineering, by looking at the whole you would start with a customer's requirements and you could carry it all the way through to the delivery of the product or the service to the customer. We felt it would apply as well to a service company, financial services, insurance services, whatever, as it would to a production or manufacturing company. So that was the basic concept under which we worked.

And we had some actual real cases where we were trying to work with this idea and we produced a series of manuals describing each of the different processes that we were working with, automated design engineering, automated manufacturing engineering, automated financial, and so forth, the whole plan. We brought in a fellow named Bob Smith, whom I had known from GE days, as a writer to help to produce this set of manuals so that we could teach people in the field how to carry out these kinds of studies. These were really quite foreign to the

way things had been done before because, again, you were looking to find a particular application, make that work in a particular part of the company. Even if you were working for accounting, you were actually working on cost accounting or on general accounting. And so the idea here was to develop a set of tools, forms and procedures that you could follow regardless of what kind of industry or what particular application area you had started with. And that was the base on which we built up that whole sequence of forms. The books still exist and they will be available at the museum showing the sequence of how a Study Organization Plan should be carried out.

LaHay: So how successful was it?

Grad: Not very. We found that it took a lot of work to design and build a major system. We ended up doing about 10 to 15 of these studies. We were able to carry them out and with certain people they were quite successful, it could work. And we'd make a significant proposal to the customer. Unfortunately, most customers weren't ready to break down the organizational barriers between engineering and planning and so forth. And so they would take a piece of it, or part of it, but they would not accept the concept that one could integrate the entire information flow in the business, and that this would be a more efficient and effective way to do it. The organizational barriers blocked it. It may have been too procedural, too detailed, but that may have only been one of the factors involved in the lack of success. After about two years we had finished the work. We had some companies that were using pieces, and said they were going to try and do the whole thing. By that point I was interested in doing some other stuff, and no one else really ever pursued the Study Organization Plan. IBM did do other business planning studies using different business planning methodologies over the years. But none of them ever really made it. Branch office sales reps and systems engineers were always busy trying to get another machine in, put some more boxes in; and doing elaborate planning studies was not what they were looking to do; it was not their favorite thing.

LaHay: So what was the timeframe when you were doing this?

Grad: My memory is this is probably 1960 to 1962 or so. Somewhere along there we decided, with IBM's agreement, that we could publish this since it was something we wanted people to use. It wasn't a matter of keeping it secret, which had been the problem at GE where they wanted to keep it in the company. IBM had a very different view. They were trying to get as many people to build major systems as they possibly could. So they encouraged us to publish. We made a contract with Holt, Rinehart and Winston, the publishing firm, and in 1968 it finally came out as a book called *Management Systems*. But by then I was totally bored with the thing. The process of producing a book for publication is just horrendous from my view. And I'll never do it again. But the book was really quite successful. It ended up that we sold some 20,000 copies. It was used in graduate schools all around the country and overseas. It was translated into Japanese and translated into Spanish, and was used as a graduate level

course in MBA programs around the country. I think something like a 100 universities eventually adopted it as part of their business curriculum.

LaHay: I even have a copy.

Grad: And I have a copy. Actually we did one follow on version some seven years later, and that was even a worse effort than the first. By that point we had to deal with online transaction processing, a whole bunch of things other than batch applications, and getting it to work, updating it, going through that travail of reading everything. We brought in two professors from the State University of New York in Buffalo to help get the book finished the first time [The title of the book was *Management* Systems and the authors were: Thomas B. Glans; Burton Grad; David Holstein; William E. Meyers; and Richard N. Schmidt]. It was just a tremendous amount of work, and I was never going to do it again.

LaHay: Let's move on to what you did next in IBM.

Decision Tables

Grad: Well, a lot of things were going on. After Bemer left IBM, maybe at the end of two years or so, I went to work for Dr. Lou Robinson. And Lou was a PhD, a fine man, and he had a scientific and technical orientation in his areas of work.

LaHay: Was this in the Research Division?

Grad: No. This was still in DPD which had a wide range of things it was doing. They encouraged me to work further on decision tables. And by this point I felt it was OK to do so. Books had started to be published on decision tables by then. And I did a preface for one of those books. I went to some of the mathematicians at IBM Research to work on the mathematics underlying decision tables because it gave you a mechanism to be able to determine whether you had a totally consistent set of conditions and actions, and whether you had complete coverage. You could check mathematically for completeness as well as consistency. I worked with people in IBM research, Ralph Gomory, Alan Hoffman, and others there. And they found this fun. They liked having real problems, and having someone with a real useful thing that could be worked with using their mathematical skills made it even better. We worked on techniques for automated programming to go directly from the tables. We worked on techniques for optimization: by knowing which conditions were most likely to occur and what states of which conditions, and by knowing which groups of conditions were most likely to occur, you could in effect rearrange the table in a sequence of performance so that you could improve the speed of the calculation 10 to 1, 50 to 1, depending on how big the table was; this would make a tremendous performance difference. Again, though, while we

found some people liked thinking in this tabular format for decision logic, we found that for other people it just didn't work for them. They were happy with flowcharts or just guessing. Decision Tables were used quite a bit in various places but never became a major useful tool. I guess it ended up being used for the next 20 years in various places. It may still be used in some places, but we never made a product out of it in IBM. In retrospect, the hardest thing was physically taking a piece of paper and writing a table, trying to create it. When you left something out or you made a mistake, you had to put new columns in. I keep thinking if we'd only had a PC with the abilities that we now have with spreadsheets, that it would have made a big difference in creating tables. I've often wondered whether it'd be worth going back and revisiting that possibility.

LaHay: I agree with you. I remember the challenges of putting together a table and saying, oh, I wish I had a column there.

Grad: I need an extra column. I used to make the columns very wide and write on one side of them so I would have extra spaces if I needed them.

LaHay: The other thing that was useful was Scotch tape so you could paper pages together.

Grad: But it was a lot of fun. Good people to work with, but it didn't become a big thing in its own right.

Linear Programming

LaHay: What about your linear programming activity?

Grad: That was an odd experience. I was working for Lou Robinson in a group that was basically interested in scientific and technical applications of computers. There were a number of usages that were called management science or operations research. Among the most significant was linear programming. This was used by the oil and gas companies particularly, but also used by agricultural firms and others. It was just really getting into heavy duty use. The concept of LP had been developed by George Dantzig in the 1950s, but the ability to solve these matrices was not easy. If you got a decent size matrix, 20 by 20, 30 by 30, you needed a pretty powerful calculation capability. If it hadn't been for computers, linear programming would never have been significant. And as we started, you could do interesting stuff with small matrices using the 1401s, and the 7000 series machines. Some companies started building linear programming codes and IBM actually had some contributed by some of the oil companies who were doing it for themselves for the 704s and 709s and so forth.

A company called C-E-I-R had a very bright guy named Bill Orchard-Hays who, with a team of people, were writing linear programming codes to run on their machines. C-E-I-R was a service bureau, and if they could have a good LP code then the oil companies would come to them and buy their machine time and they could charge premiums because they had the LP code that these other companies didn't have. And C-E-I-R had very, very ambitious growth goals. Dr. H.R.J. Robinson, the CEO of C-E-I-R, wanted the company to be the biggest service bureau in the country, focusing on technical applications. And so he ordered a whole bunch of IBM Stretch machines, I think he ordered five or six of them. But unfortunately when it came time for IBM to deliver them, he didn't have the money to pay for them. Now, IBM did not want to put C-E-I-R out of business. That would not be a good thing to do. So they were trying to work out some face saving way not to deliver the machines but to have it cost Robinson something. And so what they decided was that since C-E-I-R had a good LP code, and we needed good LP code to make available to our customers generally, not just through a proprietary service bureau we could cut a deal, and I was a negotiator for that deal.

I worked it out with Bill Orchard-Hays that C-E-I-R would produce an LP program for us to run on the 7040 and the 7044, because C-E-I-R already had an LP program for the 7090 They would do that as partial compensation for canceling the Stretch order and they would not have to pay IBM any money. And so that was the first of our own LP codes. We used that later as the basis for the LP work we were doing with S/360, and there we set up a team of 15 people to write our own linear programming codes. It happened that Orchard-Hays left C-E-I-R in the middle of this project, and he went to work for a company called Computer Applications, Inc. in New York City. And we told C-E-I-R he can't go there. You can't let it happen. He's an indentured servant. And they worked a deal with CAI so that he would continue the work and finish the 7040 and 7044 program for IBM.

One of our own people within Lou Robinson's shop did a 1620 LP code, and then for the S/360 we put together a full scale team to build the S/360 LP code, which ended up being very good. The oil companies weren't going to buy any S/360s, if we didn't have a good LP code. We weren't charging for it. It was free. We had a team of people from France, from England, from three of the oil companies, as well as our own people, to build that thing; it ended up being 15 people. We had a very small budget basically. We just sort of found the people and put it together. It was a lot of fun.

Critical Path, PERT and Simulation

LaHay: Of course, I'm particularly interested in the PERT projects.

Grad: That was another one that we did within this operations research/ management science environment. A man named Peter Norden had been doing some work on PERT within IBM, and I found out about him. The big aerospace companies said, "Hey, you got to have a

PERT program for the S/360." And so we worked it out to hire Peter to head a team of people, again with some customer people involved, to build a really top notch PERT capability for the S/360. And we wanted the similar kind of capability for smaller computers. This was called Critical Path Management, CPM. It wasn't quite as big as PERT, and we did produce our own CPM programs into late 1960s. By this point, I was pretty much working entirely on scientific and technical application programs.

LaHay: Were you still working at the contributor level, or were you a 100% into management by then?

Grad: I basically had a management responsibility, and I was not doing any programming. After I had programmed the Univac I at GE and found out that I was very fast but sloppy, that that was not the thing for me to do to earn a living. So I was managing and directing projects. Sometimes the people worked directly for me. In many cases they didn't. We had borrowed or stolen them from other groups. And so I was managing the projects. By this point I was pretty much a full time manager.

LaHay: A project manager.

Grad: A project manager, but also a negotiator and a people manager. I had multiple things going on. We weren't doing just one project. I had Pete Norden managing one project. I had another person who was managing the LP project. I wasn't managing the projects themselves. I was managing the people who were managing the projects.

LaHay: So how big was the group by now?

Grad: I didn't have a large group of people because it was mostly borrowed people. Maybe I had a half a dozen people at most as direct reports. That was all that was there.

LaHay: Including the borrowed people, how big was the group?

Grad: Oh, probably on PERT maybe 10 people. LP was probably closer to 15. Simulation was another one that was important at the time. I had done work on simulation at GE back in the 1950s. That was an area of interest to me. The idea then was to create a model of a factory and a model of the processes. We believed that we could then test various scheduling and dispatching techniques and see what happened; we could try different kinds of rules and see what occurred. At IBM a gentleman named Geoff Gordon, quite British, had been proposing an idea which was eventually called GPSS [General Purpose System Simulator]. And, again, we were able to get the sponsorship and the support needed for Geoff to produce GPSS for the S/360, to be given away to customers to help them use it for their own simulation

projects. I was not as closely involved with this project. Geoff pretty much ran that on his own. In each of these cases, I was facilitating, helping to make things happen. I didn't program. I didn't do the technical design. That wasn't my thing. I would see an application or an area or an interest, get people who had that interest together, and often with support from branch offices or from some headquarters people, or with international support, or with customer support we'd get the project staffed. You'd put together a group of people with very little money to make this happen.

Timesharing

LaHay: Did you get involved with the timesharing activity?

Grad: Only at a much later point in time. I had some involvement earlier because Lou Robinson had the responsibility for selling technical applications, and timesharing was viewed as being used for technical applications. To make this brief, IBM was beaten in that area. The first two sales I think were made by GE and maybe Burroughs, or maybe GE made both. One was to MIT, which led to the Multics project. I forget what the second one was.

LaHay: Could it have been Dartmouth?

Grad: No, I don't think so, but it might have been the University of Michigan. Anyway, the end result was that IBM management was very unhappy. We lost two big system sales because of timesharing, and so the decision was made we're not going to have this, we're not going to lose any more sales because of timesharing. And they brought in Watts Humphreys to take over that sales responsibility. Lou was moved aside. And Watts put together a team to build a product called TSS, a timesharing system, which was headquartered in Mohansic, in Putnam County in New York. And Watts didn't personally head up the development team, but there was a whole bunch of people involved, including a group of people from CUC [Computer Usage Corporation]; some of them were friends of mine: Mike Marcus and George Trimble were working on that project. And that project went on for years. And basically IBM never lost another timesharing sale. Whatever was required by the customer, Watts would promise that IBM would deliver. But TSS never quite made it. I think that somewhere around 10 customers ended up using it and IBM would have been in deep trouble except for a completely independent project that took place which eventually produced a product called VM/CMS, which was originally built in the Cambridge Science Center. I had no direct involvement at that time with it, but at a later point I did.

LaHay: What timeframe are we talking about?

Science Centers

Grad: This was the late 1960s. The sales on the timesharing systems took place, I think, from 1966 probably through 1969. TSS was supposed to be delivered in 1967 or 1968. It was designed to support the S/360 Model 67, I believe, and it just wasn't there. And when the Model 67's came out, IBM's customers said what timesharing system are we going to run on it? And that's where my connection came in. I had been working with the IBM Science Centers. As part of my assignment, I was working with Herman Goldstine by that point in time. Herman Goldstine had worked with John von Neumann at Dahlgren prior to some of von Neumann's work on the concepts of stored program machines. One of Herman's responsibilities was what was called the IBM Science Centers. At the site of a few major universities around the country, groups of IBM people were set in place, 10, 15, 20 people at each location, with very high level technical and educational skills. And the idea was to have them interface with the universities and to benefit from what people were doing there, and to pick up new ideas, and in turn to influence the universities on why IBM machines were the right ones to get, and why they should be used for their experimental work and their developmental work. And there was one Science Center in Cambridge near Harvard and MIT. There was one in New York City near Columbia University. And they worked with all the universities in the area. There was one in Palo Alto to work with Stanford, one in Los Angeles to work with UCLA, and so forth. I think there was one in Texas near Austin as well. Part of my job was to visit each of these groups, see what they were doing, and to pick up new ideas and see if there was some places where they might have commercial use, where we might be able to introduce them into the marketplace, what they later called technology transfers.

LaHay: Were these science centers all within the U.S.?

Grad: We did not, at that time, set any up internationally. But there was one set up later in Switzerland, I believe. And I don't know that there was any in other countries at that point in time. They may have been later. We worked with development groups in the other countries later on, but I don't remember any Science Centers. Goldstine did not have responsibility for any outside the U.S. Let's put it that way, because I was not visiting with any of those.

LaHay: DPD [Data Processing Division] was a completely U.S. operation, was it not?

Grad: DPD was completely U.S. And even though we worked with and had some interface with international groups on software development, it was somewhat limited at that point in time in the 1960s. So I had come across, what the Cambridge Science called CP/67 which was being developed under Norm Rasmussen in the Cambridge Science Center. And one of the ideas that occurred was that this program could be used as a timesharing system, even though it was not intended for that purpose. In effect you created a series of virtual

machines, and each virtual machine was independent. And each person had their own virtual machine, and had their own operating system, CMS; so then as far as any person was concerned he or she had their own machine. And even though they were time slicing it at the computer level, the person using the system wasn't aware of that. So CP/67 ended up being used, starting in 1968 and 1969, for some of the Model 67's that were being delivered. And one of the first ones actually was at a timesharing company called National CSS; they ended up buying one of the first Model 67's and getting a copy of what was later called VM/CMS. Even though it was not available as a product, National CSS talked IBM and the Cambridge Science Center into releasing it. They modified it and called it VP/CSS, which was their timesharing program, and they ran their timesharing system, the Model 67 using VP/CSS. An interesting set of events.

A Programming Language (APL), HASP and Program Classifications

LaHay: Were you involved with APL?

Grad: Again, later. What ended up is that the Product Divisions, who had the operating systems and the languages and had all the tools and utilities and all these kind of things at IBM, ended up being sort of single thread. I don't know if that's the right word to use. They went where they were going, and they were very long term developments, like OS/ 360, and then all the follow on programs. And the things that came out of the field, or from some individual, if you weren't part of the Product Divisions, they didn't pick up your products. So Ken Iverson produced a conceptually outstanding language called APL [A Programming Language], which was mathematically very precise, using a number of new symbols for logical operations. And the Product Divisions wouldn't pick it up. But some customers were interested in it. So we picked it up in DPD. It was one of the things that I picked up within the scientific and technical areas that I was working on.

LaHay: My recollection of APL is that it depended on VM/CMS.

Grad: Not originally. It ended up being used primarily with it, but didn't in any way depend on it. It needed an operating system environment, but it was a language, and it was then a question of where you wrote the compiler implementations for the language. And so VM was one of the places they did it because it was a highly precise tool, and the science centers liked that kind of thing. So we picked that up and supported that later on. We did the same thing with a product called HASP. HASP was a product from one of the Product Divisions. HASP was "half ASP", and again the Product Divisions wouldn't pick it up, so we picked it up. A group led by Bergstresser within DPD had done the work. A lot of work was done in the late 1960s in DPD. Remember, programs were free until the unbundling. And a lot of work was being done in the Data Processing Division with customers to support their work. The work on IMS was being done with Rockwell by an LA branch office. The work on CICS was being done

with Commonwealth Edison in Chicago. All of these things were being done in the field, and the Product Divisions were not interested in picking them up. One of my roles was making things happen, getting these things that people were doing and making them available to customers.

We had four categories of programs in the 1960s. The Type I programs were produced by the product divisions, the operating systems, languages and utilities. Type II's were being done by the Data Processing Division, which we would then support. Type III's were done by the field systems engineers. You could have them, but there was no guarantee of anything. And the Type IV's were done by customers who contributed them to the libraries, but their use was among the customers. We made no provisions of whether they were good, bad or indifferent in that regard. We were taking some of the Type III's or what would have been Type III's and trying to promote them into the Type II category because we couldn't get the Product Divisions to pick them up. A lot of these things came out of the experience of systems engineers working with customers and they became the predecessors to announced products when the unbundling takes place and IBM separately priced these programs.

Software as a Potential Business

LaHay: I think we'll get to unbundling in a bit. So at some point in time there comes the question, is software really a business? I think that question came up in IBM for quite a long time.

Grad: It did come up earlier. Remember, the IBM policies, which started with punch cards and was carried over exactly to the computers, is when you buy our equipment we support you. If you need plug boards to be plugged, we plug the boards for you if we have to. We show you how to plug them. If you need help in designing an application, laying out the punched cards and reports, we do that for you. IBM carried that model over to the computer world. And so whatever help you needed, whether it was maintenance on the machines (although in almost all cases they were leased), of course we're going to maintain them. Otherwise they'd go bad. If you needed software, we'll provide a library for you. If you can use them, God bless you. We'll support certain of them on a certain level, but others, no. They may be "selling bananas", not "eating bananas" in some cases. But we'll do that. You need education, we'll teach you. We'll run the courses. It doesn't cost you anything. We do it all. You need systems engineering help out in the field to help you design and plan what you're going to do with the computers, or why you should get a computer, absolutely. So all these things were bundled into the price of the hardware, and that was IBM's style. All the other manufacturers did it exactly the same way. It was all bundled offerings. There was nobody doing differently; they were not separately pricing, whether it was Univac or whether it was Burroughs or RCA, nobody separately priced. The man I was working for, Archie McGill, was my third level manager at that point. He felt it was a business opportunity. He was a very aggressive individual, a very strong marketing guy. He felt that there was an opportunity to

make money off the software instead of just giving it away. By that point, by the 1967 period, which is the first time that I was aware of anything being done, there may have been something earlier, but that was the first time I was aware, there were one or two companies who were selling some software. There was a company, Hankins, which was trying to sell a payroll package. There was Marty Goetz of ADR, who was trying to sell Autoflow. I think somewhere in that period, Mark IV was offered by Informatics to do application development work. And so Archie says it looks like a business. Now, I'm sure Archie had his own personal motives. He thought if they made a business out of it he'd be the head of it. That wouldn't be all bad for Archie. So he ran a first study. It was a relatively small one. And we said that there are certain programs that you could sell and make money off of. The question right away is what effect would that have on our hardware sales? Won't that hurt our hardware sales? He said, "We don't see how." I said, "Well, if we give them away, we don't lose any customers. If we sell them, we might lose some." So that was nixed the first time. Archie came back around in early or middle of 1968, with the same thing. We did a much deeper study, trying to show there were more of those programs, because by then we were starting to get some pretty good sized programs, like IMS and CICS. They were Type III's, but they were there. We saw them. We saw what kinds of major things could be done with them. And we saw some of the timesharing programs coming around.

LaHay: Getting back to the bundled stuff, do you recall the ratio of the cost of people to machines in the middle 1960s?

Grad: Depends how you measure the cost of machines to people. The actual machine costs were always very small. When you saw the IBM's pricing practices and pricing plans, the percentage of the direct cost of building a machine was probably 5% to 10% of the price they charged. And so when you talked about people, you're talking about the marketing cost. The marketing costs were 20%. But marketing included systems engineering, field engineering, as well as all your sales people. So it doesn't quite work out. It's not a simple question to answer for me, at least.

LaHay: Well, my recollection from the service bureau operations is we were renting machines at an hourly rate that exceeded the hiring rate, monthly hiring rate, for programmers.

Grad: Is that right? I don't remember because I wasn't working the service bureau side at the time, so I don't know what they were charging for machine time.

LaHay: To me, that seems to be part of the rationale for supporting the bundling argument, that the marketing costs are 20%, but if it includes all of the system engineering stuff.

Grad: It's high.

LaHay: It's high.

Grad: Of course, IBM tried to put a good sized margin on the machines. They had a lot of overhead, lot of corporate things. You had IBM Research and all these other things that were going on there. So the end result was it [the proposal to sell software] got turned down again though it was a deeper study. And I had worked on both of these. I've forgotten who I worked with on those. It wasn't a large team, maybe five or six or seven of us put together a proposal. I don't think I presented it. I think Archie did and got turned down again. Of course, it was a lot of work. There was no real proof at that point in time that software could be a profitable business. There was nobody making money off of software. Service bureaus were making money. The people doing professional services, which had become a significant business by then were making money, but nothing on the software products side. There were a couple of programs but nothing meaningful.

The Unbundling Task Force

LaHay: Tell me about the Control Data suit.

Grad: Well, again, I was not directly involved in that, but Control Data sued IBM in 1968, claiming monopoly practices, claiming "phantom machines;" these were machines that Control Data claimed that IBM announced specifically to knock off, I think it was the 6400 which was the new Control Data machine, and that IBM had announced machines that were just paper tigers and they weren't for real. So, Control Data sued. And that suit was ongoing at that point in 1968. I know from reading, but not from personal experience that at that time Tom Watson was getting advice from a variety of sources regarding IBM's liability for monopoly practices.

LaHay: Is this senior or junior?

Grad: This is junior. Senior was out as of the end of 1956 or 1957. He's gone. This is junior. And he's getting advice that the bundle won't hold. That may have been triggered by the Control Data suit or not. But he was told that you are giving different amounts of service to different customers at the same price, and that, therefore, the customers can sue you who didn't get the full service which they can claim they paid for. You could also be sued by your competitors because you were bundling. You had a tie-in sale. And by the end of 1968, by November or so, Watson was apparently convinced that he's got to cut his losses. He's got to at least protect against any future damages, and he's got to in effect block it off by unbundling, by separating the prices for the various services. He used the word "unbundle," assuming there was a bundle. So in December, 1968 Watson set up a team to look at all the different aspects of unbundling. The unbundling team was set up in what was the von Kohorn building in White Plains, a few blocks away from DPD Headquarters at 1133 Westchester Avenue, where our normal offices were. The team was set up to look at every aspect that IBM was tying in with the

sale of its equipment, and we were to come up with solutions that would separately price as many of those services as we could make an effective business case for.

The unbundling taskforce was quite a project. There were well over a 100 people working on the project and because I had worked on the business case for software previously, I was asked to serve as the DP representative on the unbundling task force for applications software. There were other DP representatives for systems engineering and field engineering in each of those areas. But I was the only person from DP on the software side. We had a very small team for software, just three people. Ted Climis was the head of the team, and Ted had come out of the product divisions and IBM Corporate, and had long term experience in the software area, and Hugh Williams was the other participant. Hugh was representing the product divisions. Ted was the head of the team, and Hugh and I represented the two sides of the software equation. Our challenge was a very simple one: we were to attack this problem on the basis that we were going to separately price software. How should we go about doing it? What products would we have to announce? How would we legally protect the products in these cases? How would we price them? We had to consider all the elements of creating a new business, but also had to keep in mind all along what effect it would that have on IBM's ability to sell hardware. How much would pricing software influence hardware sales? Would you lose sales? Would you have a negative impact on the customer's expansion and increased use of equipment? That turned out to be a six-month project, and we worked on it all through the winter of 1969. And this was an intense effort of 12 to 14 hours a day, whatever it took. Everyday we had to keep plugging away at this problem. And we had to identify all the things, all the programs that we might be able to announce.

Hugh took care of the systems software side. But it so happened that the line between systems software and applications software was very, very fuzzy. Was a database management program like IMS a system software program or an application software program? So somewhat arbitrarily the decision was made that if the Data Processing Division had done the work that it was part of DPD's decision. And if a product division had done the work, it would be their decision. Climis was a very, very strong willed person, and Ted was about six foot five, had a very loud voice, and always prepared lots of flip charts. And whenever you tried to interrupt him when he was presenting, you were in trouble. While Hugh Williams and I were both fairly strong willed people ourselves, Ted put us in the shade, for sure.

We examined the candidate programs. We came up with a number of programs that we felt were candidates to be priced. And then we had to build a business case for each of those, because the rule was very simple. If we were going to set up a business, it had to be profitable. You could not have a loss leader. You had to make money on every product which is almost impossible. And you had to make money in whatever your climate was, who was going to sell them, how you're going to price them, how you're going to support them, and how you're going to enhance and improve them. And we didn't have a lot of experience. We knew what our

costs were to produce a program. We had a pretty good idea what it would take if you really were trying to help customers use them, but we had no real good sense of what it would take to sell them, because we had never tried to sell them before. And I had never had any experience in pricing, nor had anybody else there. Pricing was very much of a black art. Pricing was totally controlled by certain financial people, and we were assigned a person to work with us from the product divisions who was experienced in pricing. And that's when I first got to see IBM's mechanism for how they decided how to price. We were told we had no choice but to follow the hardware pricing model, even though our production costs were essentially zero. Once you built the master program you could make as many copies as you wanted at essentially no cost. Their factory production models were completely different, but we had to use exactly the same models that they were using for pricing and for profit analysis. Even so, some of the programs looked like pretty clear and significant ones.

From a DPD standpoint we looked at both systems and applications programs. In the insurance industry we had two programs that had been developed in the field. One was for life insurance and one was for property and liability. Property and liability was PALIS, and ALIS was for life insurance. And we thought there was a fairly clear business case. You knew who the insurance companies were. You knew what size machine they'd have to have to use these programs. You had a pretty good idea, or you thought you did, about how many of those companies would buy them and what they would be willing to pay, that kind of thing. So you had to put together a price, using a formula to determine what the optimum price point was. But on the systems type programs, like CICS or IMS, they hadn't ever been sold. They had only a couple of people who might be interested. Were they just unique to a particular industry, or did they have broader use? We didn't know. And so we had to go through this whole pricing exercise for each of those programs. We had to put together packages as to how they would be announced. What was the announcement mechanism? What was the contract going to look like? How would you protect these programs against their being lost, strayed or stolen, because if you're going to try to make money you have to protect the asset? So that took a good part of the winter to work out all those details.

LaHay: At the end of this, how many programs did you announce?

Grad: We announced 17 programs. And, if my memory is correct, probably only a few of those were really of any significant success.

LaHay: Those being?

Grad: IMS and CICS were the big winners. And those were eventually worth billions of dollars to IBM. However, they were priced quite low to begin with and this was one of the issues that was raised after the announcement was made on June 23, 1969. We had to price them on the basis that certain customers already had these programs through the Type III

library. We had to price them on the basis that we had promised them to certain other customers on a no-fee basis, and had to deliver to them. And we had to price on the basis on what was our forecast. How many licenses did we think we were going to sell of CICS? How many did we think we were going to sell of IMS? When we came up with numbers we thought we could sell more than a hundred. The marketing people laughed at us. They would only support a forecast of, I think, 40 on CICS. You know, looking at this 15,000 copies later it clearly was a low number. And with IMS there was a similar kind of argument. We had to go through this process on each of the programs as to how many we could sell, because the optimal price point depended upon that as well as what you had invested.

LaHay: Was the original development cost treated as a sunk cost? Was it factored into the price?

Grad: The answer is mixed on that. Theoretically, that was a fixed cost, so it was gone. When you're pricing, you're pricing primarily to whatever your variable cost is. That's the way IBM's pricing formulas worked. You had to recover the cost you had put underneath it, but that was sunk cost. So the price point depended upon at what stage you thought customers would buy enough fewer copies so that your total revenue would be less. And they priced out low; if they couldn't support a forecast of more than 40, you had a fairly low, not a very high price point. As you went ahead in the business later on, of course, all your development costs had to be recovered, as well. But these were considered sunk costs that had been put into it as a Type III or a Type II program, not as a for sale program. With that announcement, you then had all these 17 priced programs.

The big decision that happened then, and that was the biggest single argument, was what do you do about operating systems? What do you do about the systems programs that the product divisions produced? They were pretty adamant that they couldn't afford to sell those and that they had to be able to continue to give them away, that the trade off in hardware and operating systems was just that, an engineering tradeoff. And therefore we drew what was called the "line," which separated between those programs which were to be priced and those which were to continue to be bundled in with the product when it was sold or licensed. IBM built in some serious problems by not immediately pricing all the operating systems. They would have been further ahead, because this mixed scheme opened the door for the leasing companies and it opened the door for the clones, because any time a customer could get the operating system free on the machine that it bought from IBM, then for its next five machines it had the operating systems. It therefore had to be locked to the machine. It got to be quite a mess. That line didn't evaporate completely until the late 1970s after which all operating system programs were being sold.

LaHay: The unbundling taskforce had over 100 people, but we've only talked about a group of three people. What else did that taskforce do?

Grad: They did everything across the board. I wasn't involved in that. They unbundled systems engineering. They unbundled all the field engineering. They unbundled education. They unbundled the professional services for major project work. They had done some of that for the government before, but now they unbundled it for everybody. If you wanted a major program done, you paid IBM f to do it. So IBM was in the full services and software products business all of a sudden, and they had to set up a whole new organization.

It required a fairly good sized organization to be set up at that point in time just to produce the software. Now here again, who was going to do the priced software? It was in the Data Processing Division. They set up a whole organization under Jim Hewitt to develop and deliver software products. We had the initial 17, which we took on, and then we were to design and build new products. As most people know, some of those new IBM services offerings worked well. But others didn't. Systems Engineering was a disaster, because the field people had lied when they were interviewed about what they did with their time. The systems engineers said they didn't do any programming. We do this, this and this, but we don't do any programming. Well, of course, it turns out that they were spending 20% or 30% of their time on programming, and when we unbundled systems engineering the customers weren't going to pay the price we were charging for systems engineers to do low level programming. That laid a great foundation for the professional services companies to really grow their businesses and to move ahead. Dave Kearns, who headed up that part of the study, the one on systems engineering, effectively lost his senior VP job in DPD as a result. He was supposed to be next in line to be the president of the Data Processing Division, and he didn't make it. He left and went to Xerox, where he eventually became the CEO of Xerox. It was interesting to see who prospered. Field Engineering turned out to be just fine. That was set up as a separate division, and it worked well and turned out to be a profitable business. Custom programming didn't make much progress, competitors were taking away much of that business. Education was a limited area. But those were not things I was directly involved in. Those were things that were happening in the company, but weren't my thing.

LaHay: How did all this relate to the Department of Justice suit?

Grad: Well, that is a fascinating little side light; because we believed that the advice to Watson was that if you go ahead and announce this on your own, you preempt a government suit. You'll still have to deal with Control Data, and maybe anybody else that comes after you, but you preempt the government suit. The problem with the government suit is triple damages. If you get convicted of monopoly, then whatever anybody wants to claim against you, if they win they get triple damages. Watson was being advised by two men who both had been senior attorneys at the Department of Justice, John Marshall and Nicholas Katzenbach, and that was their advice to him. Unfortunately, Ramsey Clark, who was then the Attorney General, didn't necessarily follow the same reasoning they would have followed, and on the last day of the Johnson administration (January 1969) he announced the anti-trust suit against IBM, which

continued for some 12 years until finally under Reagan, the government finally just canceled the suit and dropped it. I was involved a little bit in that suit as one of the people who went and did some analysis work, and that was a difficult process. So IBM had sort of the worst of both worlds. They had already given away the store and said we're going to unbundle, yet at the same time they had this suit, so everything they were doing in effect fed into the suit and said, well, you not only were doing these bad things, you recognized you were doing the bad things because you have this whole new structure in place. And so it's obvious that what we're suing you on you've already proven that we were right. So, it didn't quite play out as Watson had hoped, but that was what the original concern was.

Program Development after the Unbundling Announcement

When the unbundling taskforce was over, the Data Processing Division announced four programming industry groups. They broke up the marketing and sales into four major groups by industry. And one was financial, insurance, communications, utilities, transportation, FICUT. And I was named as the director of development for FICUT. And I worked directly with that marketing group, but I worked for Jim Hewitt, who had all the development people working for him. And there were three other development directors appointed at that same time. That was a very exciting time.

LaHay: How big where those development organizations?

Grad: They varied, but generally 100 people or more. We're talking up to four or five hundred people working for Jim Hewitt at that point. He'd also been given the programming languages, and that was another group that was working for him. It was a large group of people, and each of us had our own plans to build new products, and to deliver and support the products we had already announced, and improve them. This was going to be a business. We were going to make money, we hoped.

LaHay: Again, what's the timeframe?

Grad: It was 1969. Now, the way the announcement on the unbundling took place, was that we announced it on June 23rd, but it went into effect as of January 1st, 1970. So we had about a six-month period to get ourselves organized and be ready to be in business making money as of then. And we started working on a variety of new products and particularly on enhancing and improving the announced products. The major issue we ended up having over and over and over again was that to announce a product or even a new version of a product, we had to go through a process with the product divisions, and with almost everybody else in the company, since they had to sign off on what we were announcing. They had to agree to our business case. Our business case always was a matter of how much we were going to be able to sell. But the others had little interest in how much we would sell. They wanted to know how

much new hardware the software would sell, and that was a major problem because they would insist that we couldn't support old versions of either the computer equipment or old versions of the system software; we had to be only on the newest versions that were just coming out, so we could never be backward compatible. That makes it a little tough, because every time you start out with a product in the marketplace, you have no installed base to sell against, and that hurt our business cases over and over again and made it very difficult to announce either a new or an enhanced program product. We had to price higher, which made for a very slow, very difficult selling process.

The main thing I remember in that first year after unbundling was trying to meet the delivery dates because we had some programs already committed with dates that we couldn't meet, and we failed. And I swore at that point that I was not going to miss another date. And over the next four or five years in my development group we didn't miss one until the very end. There's a long story there which I could come to, but we put some cushion in. We didn't over promise in terms of what we could deliver. If we could put more function in toward the end, we did.

We had some very good people running the development operations. I was a director, so I had three levels reporting to me, and I had good development managers working for me, people who really understood what it meant to stay with a project. I had one person for each of the industries, one for insurance, one for financial, one for utilities, and so forth. Each of them had their own applications they were building. We were primarily to be an applications shop. That was the intention. We were going to build the applications customers would use. That turned out to be a disaster. It never worked. The big customers for the large machines would not buy packaged applications at that time. They'd buy packaged system software. They'd buy a database management system. They'd buy a communications system like CICS. They'd buy scientific and technical programs, but Equitable wasn't going to buy a packaged insurance system. They weren't going to buy ALIS. And Aetna wasn't going to buy PALIS. It just never worked. For smaller customers, when you built a program under DOS instead of OS, you had a different ballgame entirely because the smaller insurance companies couldn't afford to build their own programs, and so you could put a package out that they would buy. But we were focused for good or bad on the larger customers as far as these applications were concerned. I can't think of a major application for large companies that was successful during that period of time, at least none of the ones that I was working on. On the smaller machines, yes. We did good products for the 1130, and for the System/3. That was also part of my responsibility at that time: the applications for the System/3 and later the System/32. We covered a whole range of smaller machines. All these machines were being covered by our development groups. .

LaHay: Was that in the 1960s?

Grad: This was all in the 1970s. The unbundling takes place there in the end of the 1960s. Now you have a whole new ballgame starting in the 1970s where you're in business

now. And products like CICS made money, and IMS made money. You sold them and you sold them readily because the sales people got whole machines sold when they sold those products. But applications on the big systems didn't work well. The Linear Programming product sold well, but that wasn't one of mine. That was in the scientific and technical area. One of the other industries was scientific and technical that was set up that way.

LaHay: When did you pick up the development tools group?

Grad: You mean Application Development Methodology?

LaHay: Yes. But at some point in time did you pick up a group that was building all of these CICS and IMS tools?

Grad: I never had those. I only kept CICS for about a year and a half. A fellow named Burt Whipple was heading up that development work and it got so big that we spun it off as a separate operation, although still not in the product divisions. The product divisions did not take on the database management or data communications systems until much later. They didn't take on CICS or IMS. For years they didn't take on VM. I had picked up VM from the Cambridge Science Center. I'd picked up those other programs, like APL and HASP, which were small potatoes. DPD picked up all of those significant programs that were just lying around. It was funny, because these were really system-type programs. They had nothing to do with the particular industries that I happened to have or the other directors had. But fairly quickly Manufacturing was making its money off of IMS and not off of the applications products. Again, you couldn't sell a manufacturing control system to larger customers. It required too much customization, and IBM wasn't well positioned to customize those things at that point in time. So the standard system programs, the scientific and technical programs sold. You made money off of them, but not from the industry applications products.

So what happens? If you can't make money, you have to start reducing staff. And so within about three to four years, by 1974, if my memory serves me correctly, we had basically reduced the number of development people assigned from the four to five hundred we started with to maybe a 100 people. But all the programs that were out there still had to be maintained. And by some point, maybe 1975, I had about 40 people working for me, and by that point we had lost the System/3 that had gone to a new division called the General Systems Division, GSD. I had lost the System/7 work, to the small systems division; that included Marty Silberberg and a large number of people. You eventually cut down to a relatively small group. Most of them were just doing maintenance work, and they collapsed the organization. Instead of the four directors we started with it collapsed to basically two. I must have had 120 programs I was maintaining at that point in time. We worked up some good ways to do maintenance, to improve the processes, but the way the pricing was established and the low volumes sold, we never made money on application programs. Interestingly enough at that point in time, the

independent software vendors who had come in, the ones who were really making money, who were growing, were the ones in the database area, the competitors to IMS, and they were doing well. There were some accounting software companies like McCormack and Dodge and MSA, who were producing accounting packages and selling them. And by that point one vendor was selling a life insurance package for smaller insurance companies. There were some successful software vendors out there, but the most significant software products were the systems type programs and utility programs like Sort; for instance, Syncsort had a superb utility program. But again, IBM kept cutting back because we couldn't make money on our applications programs for large computers and large customers.

LaHay: I think programs like Mark IV and so on were around that time, too.

Grad: Mark IV and other 4GL's were selling well. And by this point utility programs were a major market. Computer Associates comes in during the early to mid 1970s with some utilities. Pansophic is in there. ADR is doing more of these things. But if you look at the dollars, Cullinet was out there. It was still Cullinane then, but it was selling IDMS, which was another database management system. So, all these system type programs were selling. ADAPSO was now a significant factor. All of those people were part of that ADAPSO. And I was there also, because I was IBM's representative to ADAPSO. And I could see how these independent software companies were producing utilities and applications. But we couldn't get them through the sign off cycle at IBM for the larger systems. Things running on the System/3 you could do. Things running on the smaller S/360 Model 30's, those running under DOS, you could get approved and released. But to get the big stuff out, we could never get them through the sign off cycle. It just didn't work.

Women as Managers in IBM

LaHay: Let me switch gears a bit, and tell me about the successful women that you've worked with.

Grad: Well, one of the things that happened during the late 1960s at IBM was they said, you've got to treat women and minorities equally. You've got to treat them the same as you would white males. And for many of us that was a new experience. You know, I thought you should treat Jews the same as you treated Christians, but I didn't think necessarily about the others. The message was very simple for women: "She's not your mother. She's not your sister. She's not your daughter. She's not your wife. She's a business associate. Give her the same challenges and the same opportunities that you give to the men. If she doesn't want to travel, she'll tell you so. Don't make the decision for her. If she doesn't want to work 12 hours a day, she'll tell you, but don't make the decision for her." And so we started to deal with it that way. And when I set up my development group, I ended up with four women managers working for me in addition to four male managers. And some of them were just absolutely terrific, and

they worked well, and it was a good experience for me. It was a whole new learning thing in working with them as managers on a business basis. And they were productive. One of them became a vice president at IBM, Joyce Wrenn, and Lucy Fjelstadt became a VP 10 or 12 years later. And the skills were there, and the management ability was there. It was just that we hadn't recognized it or accepted it before. So I found that to be one of the most rewarding things in being a manger in IBM. That was something I followed up later on with a company that I co-founded called Heights Information Technology Services which used women working at home doing programming work; in that company women managed and ran all of the projects. It was a pretty good experience for me.

Other Development Projects: System/3 and System/7

Let me add just one other thing on the earlier subject. During the first few years after unbundling, the two biggest projects that we ran besides CICS were for the System/3 and System/7. And they were completely different kinds of projects. On the System/7, IBM had announced a process control type computer that would be used hopefully by AT&T, companies like that, for operational billing and other applications which needed online feedback and analysis. The System/7 was developed in Boca Raton. Unfortunately, when the General Systems Division released it they didn't provide any operating system or any of the tools or utilities that one would expect with a product. So our challenge was to help to sell a bunch of these machines, because the minicomputer companies were eating IBM's lunch in some of these areas and IBM wanted to be competitive. So, we had to do it ourselves. I sent Marty Silberberg, one of my development managers, out to Menlo Park in California, and he headed up a group of eventually I think it was 90 to 100 people, who had to produce all the systems capabilities that one needed and then to produce applications on top of those. That was a major piece of work. Marty did a superb job. He hired well. He managed well. They produced stuff that worked. But, again, to our disappointment, the product division did not step up to its responsibilities. The System/7 was a moderate success, never a big success, but it didn't die. And the only reason it didn't die was because of all these programs that Marty and his team produced.

From a quite different angle, Jack Mumford had an idea for producing programs to run on the System/3 that could be produced automatically by specifying what the program had to do, not how it had to do it. He felt that if you wrote a set of tables and filled out a 25 to 30-page questionnaire on an application like general accounting or payroll or billing or anything like that, that he could write a program that would translate those specs into a running code. He had a 20 to 25-person team in San Jose and Joyce Wrenn was part of that team. And it actually worked. There was only one problem. The System/3 wasn't powerful enough to compile its own program. You had to do the compilation on a S/360 Model 20. Well, the System/3 customers didn't have Model 20s, so you had to do the compilation for them. And the analogy I've drawn is that when you bought punch card equipment from IBM, they showed you how to

plug your own boards. They might help you do it, but they showed you how to do it yourself. With Remington Rand equipment you had to send the plug board back to them. They had special devices to do the plugging. You couldn't do it yourself. It slowed down everything you tried to do. If you wanted to make a change, you had to wait for Remington Rand to come back with the corrected board. This was the same sort of thing with this program for the System/3. You had to go back to IBM. We really weren't in the service bureau business by that time at IBM, but you still had to come back to us. We had to rerun the compilation program with your changes, and then send the customer back the operating program. Then the customer had to run it. If something went wrong, you had to go back around the cycle again. It was just wrong. The concept was wonderful. The implementation was very well done. But the System/3, at that point, wasn't powerful enough. If we had waited a few more years, until the System/36 or System/38, it probably would have been fine. But it never got resurrected, so it died. Those were two of the most interesting projects during that period of time.

Computer Based Training

LaHay: Another thing that you worked on was computer based training.

Grad: That was fun. That was one of those accidents that happen. I'd been involved in the 1960s, when I was working with the IBM Science Centers and working with IBM Research, on something called computer assisted instruction, CAI; and I helped and worked with Ed Adams in producing the first computer assisted instruction programs which were being used inside of IBM. And after unbundling we then produced a product called ITS, Interactive Training System. That was one of the new, priced products that we produced relatively early. I believe that it came from some work that was done at an IBM branch. Later, IBM's Field Engineering Division decided that this was a pretty good thing and they started using it to train field engineers and had been working on their own instructional program, and we heard about it and picked it up. This became IIS, Interactive Instructional System, and we produced that as a product. The technical leader of the team, I can't remember his name, did the architecture for that system and he was absolutely technically brilliant. He was so much head and shoulders above almost any other system architects I'd worked with. He created an architecture that was so well structured, so functionally separated, that you could run that program on any communications system. It could run with VTAM, and VSAM, and with all the different interfaces, because he isolated all the calls. All the calls for any of the functions you needed for file management, for communications, for anything, he had isolated all those calls into separate areas so that all we had to do when we wanted to interface with something new was just change that part of the program. And all of the logic, the tools to write the courses and to run the courses were not affected at all. But besides running the courses for the students you needed a means of creating the courses. This was called Coursewriter. To write a course, you have to have the tools to create the course. But to take the course, you have to have the mechanisms

for actually taking and running it. And he split it all apart that way, just beautifully architected it. That system had a very long life and it was really quite successful.

Computer assisted instruction never became a big thing. But it kept being talked about. Control Data did work on that. It hung around for 20 years, but it never became a major way in which people learn. It was very hard to write a good course. It was very hard to write one that would really respond to the various people taking the course. At Burton Grad Associates, Inc. we tackled one project many years later as a commercial venture where we tried to build a course for Equitable Life to teach people how to handle health insurance claims. And it was just a very difficult, very hard process for someone to think that through completely and work with it. But one of the nice things that happened with this IIS work was that we started to show how these kinds of systems could be built and how the instructional elements could be built in. And we did some studies on how you would do a comprehensive system. How you'd put the whole thing together. We did one for Northern Trust, which was a bank out in Chicago, and we did one for an insurance company, INA, down in Philadelphia. And these were major studies that were run to show how you could put all the pieces together.

LaHay: This was all the training pieces?

Grad: The total training, a broad training system for the whole company, how you would do it and how you would make that work. But we followed many of the same practices in laying that out as we had done with the SOP work.

LaHay: Did they implement those projects?

Grad: Yes. In both cases they actually did follow through on that. Those were not just paper studies. People did follow through and implement the work on those. I have copies of those reports still.

LaHay: Which I'm sure you're going to share with us.

CICS and IMS

Grad: They will come to the Computer History Museum if they'll take them. I'd like to finish up on something we talked about before. CICS, which we talked about earlier turned out to be an incredibly successful program. I think some 15,000 copies were eventually licensed. It's still being used today in 2007. We did our first version in the late 1960s. Now it's almost 40 years later. I'm sure there's not one line of code that's the same as what was originally produced, but as a product it's still being sold, used, maintained, and IBM's still getting substantial license fees from the use of CICS. At one time, as a consultant, we recommended

to IBM that they put out CICS to run on PC servers, and we were laughed at. But they eventually did do it, but it was too late in a sense. They had a communications interface there that would have taken over and been dominant, we thought, in the PC world. You never know what really would have happened, but there was no really good transaction processing capability to run on those PC servers. IBM could have had it but they didn't go there.

LaHay: CICS has come through as a thread through a lot of your career. Are there some good stories about how it all got started?

Grad: Well, the stories are pretty simple. And you would get different stories from different people. One of the problems is that no one's ever done a real history of CICS. No one's ever collected the material. There haven't even been any good oral histories that I know of, of the people who designed and built the program. My first acquaintance with CICS was during the process in the unbundling task force of seeing what Type III programs might come out as Type II's, and getting acquainted with the people in Chicago. There seems to be half a dozen fathers for CICS; depending on whom you talk to, you get a different name. The people I know that I was working with were in Chicago. They were IBM SE's working with Commonwealth Edison. One of the things I'd love to see happen here at the Museum is that they would really tackle collecting those CICS oral histories and I would hope that IBM would cooperate to make that material available. But the basic story was that this was a product that worked.

One interesting, fun story about CICS and IMS that I love to tell is the following. In 1969 and 1970 each of the 15 different industries that IBM had identified had one or more online transaction processing and data base management systems that they were developing and recommending. Medical had two, government had one, manufacturing had one, and on and on and on. And we had CICS, which came out of the Utilities industry, and we also had one for each of the other four industries I had. And a taskforce was put together by Jim Hewitt to study all these 15 or more candidates and decide which ones should be in our strategic plan? Which ones should we follow? Do we follow all of them? How do we go about it? Bob Berland headed up that study, and there were representatives from each of the industries working on this. It was a fairly lengthy study, probably three to six months long. And they finally presented their results at a meeting in White Plains, and they had this mammoth set of flip charts up on the walls showing all the characteristics they were measuring for each of these 15 programs. And each of the boxes was filled in with what were the strengths and weaknesses for each of those programs. And when the presentation was done, Jim Hewitt put the basic question to Bob Berland: "Okay. What's the recommendation?" Bob's answer was: "We can't agree on anything. Each representative thinks his own is best."

So we take a break at that point and I go to the bathroom. Bill Reuther, who was the director of development for manufacturing and process industries, was at the urinal next to mine, and

made a comment about, “Hey, can we agree on something here?” And basically I said, “If you’ll support CICS to run under DOS, I’ll support IMS to run under OS.” And he agreed and we went back to the meeting and Jim said, “Are there any suggestions?” And I piped up and said, “Well, I’ll support IMS as the OS DB/DC system, if Bill supports CICS as the DB/DB system for DOS. Jim said, “Does anybody have a better suggestion?” And that was how IBM decided strategically that IMS was going to be its OS-DB/DC capability, and CICS became the DOS DB/DC transaction processing capability. It made no particular sense. CICS didn’t have a good database capability for DOS and IMS didn’t have a good communications capability for OS, but that became the strategic plan, and those were the two products that were supported going forward. And all the others disappeared.

LaHay: When IMS first came out was it primarily used for database management?

Grad: It was originally a database management system, but they had added a communications facility, just like we had added a database to CICS, but ours was weak and not very effective. Interestingly, my management people were really mad at me for giving away the OS market. I said, “Wait and see. There’s no way that they can carry the data communications for the OS market.” And within months they had approved using CICS for OS, as well. And eventually we actually buried a copy of CICS inside of IMS, because IMS communications were not good enough. The embedded version of CICS would take the IMS transaction calls and translate them into CICS calls, and we’d run CICS. But the customers didn’t know we actually had a copy of CICS hidden inside. When they bought IMS, they actually got CICS, as well.

International Development of Software

LaHay: Most of the time in DPD you were working in the U.S. At some point in time you got involved internationally.

Grad: That actually happened pretty early. Once we unbundled and we had products, we needed to have international software as well. And the concept from early on was that development should take place wherever the skills were. We should be using not just U.S. skills, but skills in Canada, skills in Europe, skills in Japan. And so I was asked to go to Europe to work with them in setting up their own software development shops. And so we ended setting up development operations in Sindelfingen in Germany (which was near Frankfurt), in Pisa in Italy, in Paris in France, and I forget the name of the place in England, and in Toronto in Canada. And we tried to set one up in Tokyo in Japan, but that one never seemed to get off the ground. Part of what we were doing after that was to come up with an integrated program development plan. What programs should be developed? Where they were going to be developed? And we set up the specs for the products on an international basis and not just a U.S. basis. And then whoever seemed to be most appropriate would do the implementation on a worldwide basis. In IBM, documentation was fundamentally in English. And though you'd

have to have local versions for the people who were buying the programs and using the programs, that was essentially a translation from English. It wasn't the easiest thing to do in France, and Germany and Italy, but that was the concept. They always had to produce an English language manual, an English language spec.

I was going over to Europe maybe two, three, four times a year. A fellow named Rainer Kogan would go over with me. He spoke both German and French, but not much Italian. And he would go with me as my aide on these things to help me in two ways: one, to translate if there were language problems, but the other two was to drink for me. I was never much of a drinker. And in France, in Germany, and in Italy if you didn't drink with them they felt you were trying to put one over on them. That was somewhat true in England, but not quite as much. And so one of his roles was to sit beside me and get my glasses finished so I didn't look like I wasn't drinking. Rainer was also very competent technically. He had been in Germany, had actually escaped from one of the prison camps. His mother was Jewish and had been sent to one of the concentration camps.

The headquarters for the European software operations was set up in Amsterdam. And so we would often meet there, and we would actually go to each of the individual locations at least once or twice a year. That was quite an experience. The results were so-so to poor. Some of the groups were fairly good at producing packaged programs, products that would really be valuable. But there were some serious problems because of the difference in cultures. For example, let's look at the French view as an illustration. The man who would do the architecture and the basic system design was a top flight person, a really sharp guy. I remember the work on the critical path method program for the smaller machines, where he did a superb job. And then as they started implementing the program, the stuff that was coming over for us to look at was third rate. And when I talked to him about it I said, "You're a good programmer. What are you doing?" He said, "Well, I don't write the program. I have these trainees do that. That would be below me." I said, "But they're doing a terrible job." He'd say, "Well, that's the way it is. This is what we do." And this happened worse in France than in the other countries; it happened to some extent in Germany. Italy never quite stepped up to the plate. The Brits were pretty good. They did some nice work. At a later time they took over CICS and did a good job of maintaining and supporting it.

The other thing that was difficult was once you agreed upon a spec they said, "That's locked. This is the way things are." It's the same way they did their financial plans for the countries. You know, France had their five-year plans like the Russians did. We said, "Well, something's changed. Online transaction processing is what you need. You need to be using terminals. You can't just have batch processing for the critical path program." Their response was, "This is what you agreed to. This is what we're going to implement." "But we can't sell any if you give me that." "That's what you agreed to, that's what we're going implement." The end result was whatever they gave us we had to basically throw out. We couldn't sell them in the United

States. We couldn't make money off of them. And that was a damn shame, because there were some very talented, very competent people in the European software centers. And yet those cultural styles in the 1970s just didn't work in what was a very rapidly-changing world. The online transaction processing world had happened to us. Everything had to be using database. Everything had to be using CICS. By that point CICS had essentially become a standard, not just in IBM but across the country and the world. These other software companies could sell against IMS, and win. Maybe it was because of IBM's potent marketing and not because the product was so good, I just don't know. But the end result was that CICS became the standard. Everybody who wrote applications wrote against CICS.

Completing Development Work at DPD

LaHay: Towards the end of your career in IBM you moved from DPD to the IBM Research division. First of all, why?

Grad: Because I was kicked out of DPD. <laughs> It was very easy.

LaHay: Is this because of your outstanding success in delivering application products?

Grad: You got it. No, the big thing was that I missed one delivery date. Two things had happened. One, as the size of the development operation collapsed they didn't need four development directors anymore. And so I was then reporting to one of the development directors, to a fellow named Stan Alekna who was an interesting man. His technical knowledge was limited. But he was a good marketing guy with a very strong personality. For fun he used to take a sledgehammer and break up rocks. That was what he thought was fun. He and I went out to the West Coast one time to visit my development groups there, and both of us decided that we had to develop a relationship. So we decided to play tennis. He thought he was pretty good. At the end of the fourth set, when I had won 6-0 for the fourth time, he decided maybe he should be my partner in the future and not play against me. But on the way back to the east coast, we stood in the back of an airplane for almost the entire flight arguing about what we should do in terms of future development work. This must have been around 1975, somewhere there. We argued and argued, and we finally came to an accommodation. He and I got along great after that. And he was supportive, and very helpful in getting messages up to senior management, helping us get money when we needed it, that kind of thing.

But Joe Henson and Bill Almon who were Stan Alekna's bosses told me that they needed a newspaper publishing system for the 1130. And I said, "Fine. We'll work on that." That was part of one of my industries then. I had almost all the industries by that point, except for distribution, banking and scientific. I had all the others. And I said, "Fine. I'll put a staff together." "No, no. There's a guy in Cleveland who is going to head this up, because he did it before on a previous machine, and he'll do a great job." I said, "Well, I'll go meet him." And he was an unusual man.

He was sort of half-crazy, very self-motivated which is good sometimes, but he wouldn't listen to anyone. I said, "I can't use him, but he'd be a good consultant to the team. I'd love to have him. But he can't run the project; he's not a project manager." "He's going to be your project manager." I said, "He's in a branch office." "He's going to be your project manager." I still don't know why they insisted. So I said, "Okay, we'll run it," and I put one of my people on the project who was good technically to be sure I knew what was going on and we could manage it from a project standpoint. Then I said, "Here are the dates I can meet. But management said, "No, we're going to lose these newspaper orders if you don't make it six months earlier." I said, "I can't do it. Not with him. Get me somebody else, let it be my team, and I'll meet your date." But the answer was no. He missed the date. That was the first delivery date that I'd missed in, I think, four or five years. I had over 100 deliveries during that period and had not missed a date. But management said, "Bad management." I said, "Okay. We'll do it again. Here's a new plan, here's the date I'm going to make." "That date's not good enough. It's got to be six months earlier." I said, "Okay, get rid of the guy in Cleveland, and I'll meet that date." "Nope, you've got to keep him." And he missed that date.

By that point I felt that the writing was on the wall. I was not happy with the support I was getting. That was a "failure" which they could point to. And I said, "This isn't working. I'm not ecstatic about not being a development director any more. Let's find me something new to do." I then had some discussions about what to do. And Joe Henson, who was the VP of industry marketing and I talked. And he always called me Dr. Grad for no good reason. I said, "I'd like to do something more interesting." We worked out that I'd go work at IBM Research and I would see what things they were doing there. And I would try to transfer some of that technology, and get products out of it, get some of these technologies moved into DPD and the product divisions and get them to be useful products. And that was my goal in going up there. And I thought I'd be there a year. DPD agreed to pay my salary for the year.

IBM Research

I worked for Joel Birnbaum who was a superb researcher and leader of the people at IBM Research in Yorktown Heights, N.Y. The people were doing some great stuff. And I ran across a friend of mine up there, Harry Markowitz. And Harry Markowitz is the only Nobel Prize winner that I know. He later on got the Nobel Prize in economics for the work he had done back in the 1950s on portfolio selection. And I was able to get him the support he needed at Research [to work on EAS-E, a follow on idea to Simscript] and was able to work with these other PhD's to identify some technologies that should get transferred. But I was not successful at getting them transferred. Among the projects that Birnbaum had at IBM Research were the RISC machines being designed by John Cocke, and Birnbaum was very knowledgeable in working with the relational database management system concepts that were being worked on in San Jose and Santa Theresa. SQL was designed there using the mathematical concepts that Ted Codd had developed [and System R was being implemented]. And all of these things and others were

sitting there available to be transferred. But I couldn't find any way to get the product divisions to look at these products or to accept them, and we couldn't move them into the Data Processing Division because they just didn't fit well there. It was very disappointing.

The other assignment that I had a lot of fun with at IBM Research was on application development methodology. I had always been interested in that subject. I had done some work in DPD before and developed a product which was called the Development Management System. We had it running under CICS. While at Research I then surveyed everything going on in IBM and outside IBM in development methodology systems. There was a mammoth amount of stuff going on. All these were tools that essentially tried to get you away from procedural programming, from having to write a detailed procedure, and instead to try to describe what it is you wanted the result to be, describe your inputs, describe your transformations and let the development system put those pieces together. The idea was to give the users tools to write good application programs cheaply so we could expand the equipment base. During the 1970s the base of usage wasn't expanding anywhere near as fast as we'd hoped, and the technology was moving so fast on price/performance for the hardware, that the new machines IBM was delivering to you would have three times the power for the same dollars; but you didn't need a lot more power because you didn't have the additional applications to run on them. This was the genesis of what became an inside IBM joke; they didn't tell me to make the programs run slower, but they didn't tell me to make them run faster either.

We were trying to develop tools to improve the ability to produce programs quicker and cheaper. This effort produced a very valuable result, but I don't know how to measure what the total effect was. At that point in time I'm not sure IBM did much more as far as application development methodology was concerned. But there were dozens of independent vendors out there who ended up building various development management systems, ways to build applications faster and cheaper than they had before.

I was trying to bridge that gap between Research and the product divisions or the marketing divisions so that we could have products come out of the research activities. But it didn't work well. There was a man named Moshe Zloof at IBM Research who built a program called Query By Example [QBE, based on the Codd relational model]. And he had a number of customers who were actually using the program. But we couldn't get it announced as a product, couldn't get any of the other divisions to take it on as their product. Of course that's to some extent what happened with relational database systems in IBM. Santa Theresa had a program, System R, running there in the mid 1970s. How good it was I can't answer. But IBM, because they were afraid of impacting IMS, wouldn't push that ahead. We have stories on that from previous oral history interviews and workshops, and that's very disappointing that IBM missed out. And they missed out on the RISC machines. IBM came in later, but Hewlett Packard took leadership in that area when Joel Birnbaum joined them.

So those were areas where I was trying to help to bridge that technology gap and I was unsuccessful. And this may be fairly typical of research centers, I just don't know. The research center inside of a company should be producing things that get used within the company. But you look at Xerox PARC and all the wonderful stuff they produced, and yet Xerox never was able to take any advantage of it. AT&T took some advantage of the transistor concepts that were developed at the Bell Labs. They did take advantage of that, but slowly. So there may be some inherent problem with technology transfer within most companies. 3M seems to be the opposite side of that coin. They seem to have a magnificent track record of taking things being developed and designed in a lab setting, and getting it marketed. Look at Post-Its. It is one example that always comes to my mind of how they developed a glue, which by accident didn't stick very well. It kept coming off. And so they came up with the brilliant idea, oh, wouldn't that be nice. Something that sticks but then doesn't stick. But that didn't succeed at IBM Research.

LaHay: One of the themes that I hear all the way through is the kind of silo of the product divisions.

Grad: Absolutely. And if it wasn't for the Data Processing Division in the 1960s and 1970s, the major IBM impact of CICS and IMS would never have been in the marketplace. They never would have happened. And that was worth enormous revenues to IBM. The amount of revenue for CICS software alone, forgetting about the tools, forgetting about the hardware drag, was tens of billions of dollars. And so that lack of acceptance by the product divisions was very disappointing to me.

Leaving IBM

At that point in time I decided I'd overstayed my time in IBM Research. I was supposed to be there for a year, but it was actually 18 months. I had nothing really to come back to in DP. I'm not even sure I would have been welcome back there. And so the question was, where do I go next? What do I do next? But I didn't find anything in IBM that I really wanted to do at that point in time that would have been fun. They offered to have me to go run the Science Center in Cambridge. That would have been fine, but it wasn't in the mainstream. And I felt like somehow I wanted to be in the mainstream of what the major things were that were going to happen in the computing world; and I didn't think that the Science Center at that point was going to be a mainstream location. And so I made the decision to leave and take a risk. And as one final foray I went to Russia for IBM to see if there was a way of sharing software with them, which there wasn't.

I'm not a real entrepreneurial type, but I decided I would start my own consulting business. By then I had known the ADAPSO software people so I knew all of these software companies, the independent software vendors and I figured I could get some work consulting work with them which I did. That takes me up to the end of January, 1978 when I left IBM. And I believe I left

IBM with good feelings on both sides because IBM used me as a consultant for a couple of projects after that, and they were good fun.