

An Interview with  
HARRY M. MARKOWITZ  
OH 333

Conducted by Jeffrey R. Yost

On

18 March, 2002

San Diego, California

Charles Babbage Institute  
Center for the History of Information Technology  
University of Minnesota, Minneapolis  
Copyright 2002, Charles Babbage Institute

## Harry M. Markowitz Interview

18 March, 2002

Oral History

### Abstract

Harry M. Markowitz discusses his development of portfolio theory, (for which he received a Nobel Prize in Economics), “Sparse Matrices,” and his work at the RAND Coporation, GE, CACI, and IBM on simulation software development, modeling, and operations research.

Yost: It is March 18, 2002. I'm Jeffrey Yost of the Charles Babbage Institute and I am here today with Harry Markowitz at his office in San Diego, California. This interview is part of the Charles Babbage Institute's NSF-sponsored project "Building a Future for Software History." Today I will primarily be concentrating on software development and the early software industry. I, however, am also interested in getting some context on your evolving thought that led you into the software field. Could you please begin by briefly describing how you became interested in economics at the University of Chicago?

Markowitz: That wasn't a longstanding decision on my part. I went to the University of Chicago that had survey courses that led to a Bachelor's degree, called a Bachelor of Philosophy. I took placement exams before going into the program and I got excused from the physical sciences. I did not have to take the survey courses of the physical sciences because I had read enough in high school on my own to get excused from that. So when I got through with the survey courses, when I got my Bachelors, I had to choose which department I was going to go into. The physical sciences were not in my head anymore at that time. I had enjoyed math. And I had done reading in the social sciences as part of one of the survey courses. I liked the applications that economics had; the theoretical structure to the discipline. I just decided, okay, I've got to make a choice. I will choose economics. When I was in high school I read a lot in philosophy. I had read some science, more or less at a popular level, but I read the philosophers themselves, including Hume, who struck me as having something very interesting to say. I was especially interested in "what do we know," and "how do we know it," and the "uncertainty of it all." So the part of economics that interested me the most was the "economics of uncertainty," particularly the "theory of

games” and utility theory with von Neumann; Morgenstern; Friedman and Savage’s “Utility Functions.” That was the area that attracted me within economics.

Yost: How did you get interested in research on portfolio theory?

Markowitz: Now several years later, I am at the stage where I have to choose a dissertation. I am now at a Masters and I am working towards my Ph.D. I went to my advisor, Professor Jacob Marschak, to ask him if he had any suggestions about a dissertation topic. He was busy, so I sat out in his anteroom. There was another gentleman there and we got to talking. He was a broker and suggested that I apply mathematical statistical techniques to the stock market. So when I got in to see Professor Marschak I said, “The guy out there suggested I do a dissertation on the stock market.” At the time I was a student member of the Cowles Commission and Marschak had been formerly the head of the Cowles Commission. Marschak explained that Alfred Cowles, who had endowed the Cowles Commission, was particularly interested in the application of econometric techniques to the stock market. Marschak did not know the financial literature, and he suggested I see Professor Marshall Ketchum in the Business School. Ketchum gave me a reading list including Graham and Dodd; Wiesenberger, on investment companies and their portfolios; and John Burr Williams’ *Theory of Investment Value*. He [Williams] was really the theorist of the day, sometime in 1950 this was. Incidentally, March 1952 is the 50<sup>th</sup> Anniversary of the publication of my “Portfolio Selection,” sort of a special anniversary for me. But anyway, sometime in 1950 I was reading John Burr Williams in the Business School Library at the University of Chicago. Williams stated that, “The value of a stock is the expected present value of its future dividends. I thought to myself, well, if you are only interested in the expected value

of a stock, you must be only interested in the expected value of a portfolio. And the expected value of the portfolio is maximized if you put all your money into one stock, the stock that has the maximum expected value. I knew that wasn't the way people invested because, of course . . .

Yost: It is not very safe.

Markowitz: No, it is not very safe. I had read Wiesenberger and he had shown all of these diversified portfolios. Obviously people diversify to reduce risk. The first thing I thought of as a measure of risk was standard deviation, because that was the most common measure of dispersion that the statisticians used. I thought of the return on the portfolio as being a weighted average of the returns on the individual securities where you chose the weights. I knew what the expected value of a weighted sum was, but I didn't know what the variance of a weighted sum was. So I took from the library a volume, that of [J. V.] Uspensky, *Introduction to Mathematical Probability*, and looked up the formula for the variance of a weighted sum. It had all these covariances in it. That seemed just wonderful because the variance of a portfolio clearly depended not only on the variability of the individual holdings, but also to what extent they went up and down together. I was a budding young economist. I had two things, risk and return, so I drew a diagram with risk on one axis, return on the other axis. I drew a little figure, which we now call the "efficient frontier." That was the world's first "efficient frontier." There are obviously points on the frontier and points off the frontier. At the time I was taking Tjalling Koopmans' course on activity analysis. Later Koopmans would get a Nobel Prize for this work, but this was before Nobel prizes [in Economics] were given out. Koopmans distinguished between efficient

and inefficient allocations of resources. Efficient ones were ones where you couldn't get more of one thing without giving up something of something else. So I spoke about efficient and inefficient portfolios, and that afternoon, that much of "Portfolio Theory" was developed. That still left things to do. Like developing a computing algorithm. So as part of my dissertation I started working on this computing algorithm.

Yost: Had you used computers at this point?

Markowitz: Well, at that time, computing technology, as far as we had it at the University of Chicago, consisted of, I think they were called Marchant Machines.

Yost: Yes.

Markowitz: They were fancy calculators. I hadn't worked out what I later called the "Critical Line Algorithm." It turns out the set of efficient portfolios is piece-wise linear. All you have to do to get every point on the entire efficient frontier, is to figure out which are the corner portfolios. These corners happen where a security drops in or a security drops out as you move down the frontier. Some linear equations have to be solved and you keep updating an inverse. But I didn't work that all out until after I left Chicago. I guess I was working on it as part of my dissertation. I left Chicago in 1951, with everything done except the dissertation. In 1952, I published "Portfolio Selection" in *The Journal of Finance*. The article argued why you wanted to believe in mean and variance, rather than just expected return. It illustrated what efficient sets looked like with three security examples. But I didn't publish the Critical Line Algorithm, how

to actually compute these things for large numbers of securities, until about 1955 or 1956. I published this in the *Naval Logistics Research Quarterly*, which was publishing a lot of algorithmic stuff at that time because Alan Hoffman was editor. Incidentally, he received two quadratic programming papers roughly at the same time: one from me and one from Phil Wolfe. He sent mine to Phil Wolfe and Phil Wolfe's to me, and we both recommended publication. Anyway, later I was invited by James Tobin, who just died recently by the way, to come--I was at the RAND Corporation-- spend an academic year at the Cowles Foundation, which had moved to Yale, and to transform my dissertation into a book.

Yost: That was in 1956?

Markowitz: Yes, 1956. Right. At the Cowles Foundation we didn't have the fancy kind of internally programmed computers like we had at RAND, like the JOHNNIAC. We had something where you had to wire a board. I can't remember its name now, but it would go through sixteen program steps and what it would do on each program step depended on the wiring, and so on. By successive runs through that computer, I was able to do the "efficient frontier" that appears in my 1959 book.

Yost: Can you go back a little bit and talk about your decision to go to RAND. Was it to work on linear programming?

Markowitz: Life marches on. We talked about being in high school, and the early days at the university, the later days at the university, now I had come to the point where I had to pick a job.

I got an offer with decent salary from some university. I asked the head of my department what he thought of the offer. He said, "Harry, don't sell yourself short." I figured, he thought I could get a better offer. Then at the American Economic Association Meeting I met the RAND folks from the economics department and told them about my interest in Portfolio Theory. Maybe they had already heard about it. What we now call "Operations Research Under Uncertainty" and so on, was clearly right down RAND's alley. So they made me a very attractive offer. It was fifty percent again as much as the university [offer]. So we accepted their offer. And of course they are in California [Santa Monica]. I already had a child and the other one was on the way when we accepted their offer. I remember watching the weather on the television. We had a horrible winter. I think it was 19° below in Chicago. Chicago usually doesn't get down much below 0°, but that's how really horrible it was that year. Then we saw it was 60° or 65° in California and we'd cheer. When I got to RAND, after I had been there a few months, a small group of economists, three or four of them, came to me and said they were doing some linear programming and they'd like me to help them. They'd like me to read an article by George Dantzig on the "Simplex Algorithm" and supervise the programming of his algorithm. Even though this algorithm was published in a Cowles publication later, I had not read it before. I knew about the Kuhn-Tucker Theorem and I had used that, but I had not read about the Simplex Algorithm. So I read the Simplex Algorithm. Needless to say, George Dantzig was not there yet—obviously, otherwise, why bother with me? This may be of some historical interest, Cliff Shaw, the "Shaw" of subsequent "Newell (Allen), Simon (Herbert A.), and Shaw" artificial intelligence fame, did the programming. I think we were on something called a card programmed calculator. This was 1951 or something like that. I don't know whether the JOHNNIAC was around yet. We got up to four [simplex] iterations a day. About a year later

Dantzig came and continued to revolutionize linear programming at the RAND Corporation. It was of great personal value to me.

Yost: Can you talk a little bit about your coordination of LP/I for the Air Force in the late 1950s?

Markowitz: Sure. That skips over one thing of computational interest. Can I go back? Can I backtrack a little bit?

Yost: Please do, are you referring to Sparse Matrices?

Markowitz: Yes, Sparse Matrices, right, because by that time I am out of Sparse Matrices, I'm onto something else. Alan Mann and I engaged in a project with other people, not necessarily from RAND. There was something that was around at that time called a Leontief input/output matrix. He [Leontief] tried to forecast the direct and indirect effects of final industrial output. He tried to predict this with a square matrix where each row and column represented an industry; and the entries represented industry inputs per unit of industry output. Alan Mann and I had examined it and we had found some strange consequences of this input/output matrix. Eventually, we wrote a book about some of these strange consequences. We proposed as an alternate, building linear programming models of industries with the same--they would be fairly aggregate models--objectives. We called these models "process analysis." Our objective was to produce a more sophisticated way of understanding economic capacity. When we started putting these models together they became very large as compared to linear programming capability at that time. A general linear programming algorithm at that time would take about two hundred

equations. Special problems had greater capacity like the Ford-Fulkerson Algorithm on maximum flow problems, and so on. But just for a general-purpose linear program, two-hundred equations was max. It struck me that our matrices were mostly full of zeros, and if you have a set of simultaneous equations that are mostly zeros, if you pick your pivots right, you could just solve it by hand. Then I thought, well, maybe we could get the computer to do the same thing. This led to "Sparse Matrices." As far as I know, I coined the word "Sparse Matrix." What wasn't very sophisticated was our bookkeeping procedures for keeping the non-zero coefficients around in an efficient way. Others have done a lot of subsequent work on this. Bill Orchard-Hayes programmed my algorithm and then decided, boy, that was the hardest thing he had to program. He'd never do that again. I published [in this area] and then forgot about it, until I was visiting IBM Research many years later. They had just had their second conference on Sparse Matrices. So it had a history of its own.

Yost: Did it take hold quickly after you published this research?

Markowitz: I published it in 1956 or 1957. This business of seeing Alan Hoffman while visiting IBM Research, that was in early 70s. So there was a decade in there. Somehow during that decade, it started to catch on and expanded very fast. I was completely oblivious to was going on and to what extent.

Yost: Could we return to my earlier question about your coordination of LP/I for the Air Force in the late 1950s?

Markowitz: I was in the [RAND] Economics Department when I did this research. It was in a sub-department within Economics, called Logistics. The Logistics Department put together--this was somebody else's conception--a big lab experiment to simulate Air Force flying missions on a computer—addressing malfunctions, and keeping track of logistics stuff, the supplies and maintenance and so on. The plan was to have a controlled experiment where you compared old-fashioned inventory control, and new-fashioned inventory control techniques, and saw how they worked in this big man/machine simulation. My function was to coordinate the development of the computer part. The computer was simulating this thing. This was at a time when tapes were the mass storage. There were disks around, but back then we didn't have lots of them around RAND. I had a team of four or five very sharp programmers: Jim Tupac was on the team. Jack Little was on the team. I can't remember the other guys. One of my major functions was to first gather from the Air Force types the specs that they wanted to see in the models we produced. Then secondly, help kibitz the big design, how we would go about it. Maybe one of my most important functions, once we started programming, was telling the Air Force guys, "We cannot make any changes at this time. Before we set this into concrete, we will review all of the proposed changes. In the meantime, I will keep a list of changes that people want. When the thing was working fairly well, we had a big meeting. I said, "I've got these twenty-some proposals that people want. We will all discuss them and decide." We discussed proposal number one, and one guy was for it, and everybody else didn't think it was worth changing. Maybe out of the list there were two or three things that everybody decided, yes, that had to be done. So we did those things. We got it ready almost on schedule. By the way, this wasn't my first exposure to simulation.

Yost: Can you give some background on your earlier exposure to simulation?.

Markowitz: Sure. When I was in high school I developed an interest in philosophy. I read philosophers addressing questions such as “What do we know,” and “How do we know it.” In college, when I got to economics, it was “Economics of Uncertainty” that appealed to me. When I did my dissertation it was “Portfolio Theory.” But I didn’t consider myself particularly a finance guy. I don’t know if the word was around, but I considered myself more an Operations Research kind of guy who was particularly interested in applying mathematical or computer techniques to real life practical problems, particularly when they deal with randomness or uncertainty.

Yost: Yes.

Markowitz: I had been meeting with a group over at UCLA under Mel Salveson who was trying to apply various advanced techniques to manufacturing problems. It was clear that you needed to do simulation, you couldn’t optimize. I mentioned before, this process analysis thing, we got various people interested in doing models of various industries. Alan Rowe and I did a model of the metalworking industry. So I got to know Alan Rowe who was at that time an industrial engineer working with UCLA, and who was on the Salveson Project. Also, at RAND there were large warfare research projects--including air war simulations--where enemies would be invading, the bombers coming in, and the fighters, and so on and so forth.

Yost: And at some point you continued this type of simulation work at General Electric?

Markowitz: When I was at the RAND Logistics Department, I got an offer from General Electric I couldn't refuse. The Computer Department of General Electric wasn't a great place to work; it wasn't a great computer department. But I then got an offer to come to General Electric Manufacturing Services in New York. The Computer Department was in Phoenix. The Manufacturing Services Department of General Electric serviced the various manufacturing departments.

Yost: Do you remember what year this was?

Markowitz: This was roughly 1959. I'm a little fuzzy about the exact date. Eventually, I went back to RAND. I know the first SIMSCRIPT was published in 1962, and so I must have been there at least two years. So it was two years prior to that, that I was at General Electric. Alan Rowe, who I had worked with on the process analysis stuff, had moved to General Electric. He had built a big, what he had hoped would be a very flexible job shop model--a realistic one with all the bells and whistles for somebody at their job shop. It was programmed, not by him, but by a programmer under him, in assembly language. It turned out it wasn't as flexible as he had hoped. He had a lot of parameters that could be set to tailor it, but somehow when the next big realistic job shop presented itself there were features of that job shop that required reprogramming, not just parameter setting. The model wasn't really suited to that. So at that time, I had a theory. I became enamored of FORTRAN [*FORM*ula *TRAN*slation]. I think FORTRAN I was out at that time. I am not quiet sure when FORTRAN II came in, but maybe FORTRAN I ,was there.

Yost: Yes, I think that FORTRAN had been out for two or three years.

Markowitz: Yes. Anyway, I had read [the manual] and I thought FORTRAN I was wonderful. You didn't have to talk to programmers; you could program yourself. Those were the days when programming manuals were less than one hundred pages. I had a theory that what we could do is make reusable modules, putting them in FORTRAN. A GE department presented itself and wanted simulation. Mort Allen did the programming. We thought about the program and built this General Electric manufacturing simulator. That worked just fine for the first department. We gave lessons; we'd have seminars within General Electric about how you should use manufacturing simulators. We called this particular manufacturing program GEMS (*General Electric Manufacturing Simulator*), and we thought GEMS could have a lot of applicability. I heard years later, after I left General Electric, that there were at least a few die-hards in there using GEMS. But it turned out that it didn't have all that much flexibility, and these reusable packages weren't all that reusable, except for a few that did things (like in SIMSCRIPT later), created entities, filed things into sets, kept an event timer and so on. So a new theory evolved. What we needed basically was a language that would do this: put at the programmers disposal the ability to create and destroy entities, et cetera. I decided that if I built that within General Electric, at least within Manufacturing Services, it would be likely to just be used internally.

Yost: It would be considered proprietary?

Markowitz: Yes, proprietary. I wanted to find a nice RAND-like environment to build this thing. The only two places that really qualified that I got offers from--I don't know whether I tried contacting Bell Labs, that would have been the third place in the world that I might have considered--were IBM Research and RAND Corporation. It was hard to choose, but I decided, okay, RAND, go back to RAND. The word SIMSCRIPT wasn't in my head. But one day Herb Karr and I went up to a blackboard and thought of all the ways of naming a simulation language and somehow SIMSCRIPT came out of that.

Yost: Please provide the background of your early work with Simscript: You worked on it with Herb Karr and Bernie Hausner?

Markowitz: Yes, when I came back to RAND I said, "I want to build this simulation system." So they assigned me a programmer, Bernie Hausner, who is very, very smart. He and I worked together and puzzled out how sets should be held together and so on. In the first instance we were developing a preprocessor to FORTRAN. You had to put a special mark on your FORTRAN pad to indicate "this is a preprocessor statement." The thing that we were developing we called SPS-I, not to be confused with something else called SPS. Simulation Programming System (SPS-I) is what it was called. Herb Karr called up. Now, I knew Herb from General Electric days. He was sort of a wheeler-dealer, and I don't mean that as necessarily derogatory, but he was more interested in making money than producing theorems, which is fine. There is an important role in life for people like that. When I was at General Electric, consulting for the Computer Department, he was at General Electric TEMPO [*TEchnical Military Planning Operation*] in Santa Barbara. I was a consultant to him. He had some other operations research

types, RAND types, consulting for him also. It was a fun group. He later moved to Planning Research. I had moved back to RAND. He moved to Planning Research, and he got fired. He was looking for a job. I said, "Would you be willing to write a programming manual?" He said, "Sure, I need a job." So SPS-I was done, and we were now working on SPS-II, later called SIMSCRIPT. We asked him to write a programming manual for SPS-I. He said, "By the time I finish the programming manual, SPS-II will be ready. Let's just write a programming manual for SPS-II." Then we did the little exercise where we renamed it SIMSCRIPT. The result was a collaboration between Bernie Hauser, who did the programming, he programmed the preprocessor into FORTRAN; and Herb Karr who wrote the manual; and me. What was I doing? The three of us would meet together a couple of times a week and work out the fine details of the language. I guess when I wasn't helping design the fine details of the language maybe I was reading a math book. But anyway, SIMSCRIPT I came out.

Yost: Were SDC's SIMPAC and IBM's GPSS ( *General-Purpose Systems Simulator*) projects going on concurrently to the development of SIMSCRIPT?

Markowitz: Yes. We didn't know about--what was the first? Tell me the first one?

Yost: SIMPAC.

Markowitz: I don't know. SIMPAC never got to my consciousness. When we were building SIMSCRIPT, I wasn't aware of other work going on. I had come from General Electric and we had this problem with the inflexibility of the simulators. I had this theory, and I got a

programmer, and we just worked at it. I didn't know about GPSS or GASP or--there's another one that I can't remember—until after SIMSCRIPT was published. Then Herb Karr said, “Why don't we go into business together?” He'd been saying that for years, ‘Why don't we go into business together.’ This time I said, “Sure. Why not? It can't be harder than writing a compiler.” We founded CACI. We each put in one thousand bucks and later--maybe a few months later--each put in an additional five thousand bucks. That was the initial capitalization.

Yost: Was there any outside financing?

Markowitz: No, no outside funds, just us. What we did in the first instance was, two kinds of things. One was we tried to stay off of CACI's payroll by working elsewhere. I consulted at RAND, and Herb consulted at Douglas. Then we conducted SIMSCRIPT courses. We got a mailing list from an appropriate place and sent out ads. These courses were very valuable. It made us a little money and it made us contacts. Some of those contacts became sales and some of those contacts became people we hired--so we bootstrapped from our contacts.

Yost: So SHARE was distributing SIMSCRIPT, but you were providing services on how to use it, particularly applications?

Markowitz: Yes. Right. RAND put SIMSCRIPT into SHARE and we were just providing education on how to use it and consulting on how to use it. We gave a special course to the Air Force, and consulted for the Navy. I am a little vague on some of these details.

Yost: Can you remember any of the other early customers?

Markowitz: Well, mostly we gave these courses. People came from all over. I'm uncertain of the specifics. Amazing what you remember and don't remember. Anyway, there were two important things...

[Break]

Tape 2: side A

Yost: You were speaking of two important things that happened in the early years of CACI?

Markowitz: One was IBM wanted to have a version of SIMSCRIPT for a new operating system. But first I have to back up and tell you what was happening with SIMSCRIPT II, even though the thing that we were about to produce for IBM was not SIMSCRIPT II, but something we called SIMSCRIPT I.5. It used compiler technology that I had developed for SIMSCRIPT II. There were various things we didn't like about SIMSCRIPT I, or opportunities that we felt we could take advantage of in building SIMSCRIPT II. We didn't like going into FORTRAN. We wanted to go to assembly language. Also, SIMSCRIPT I and SIMSCRIPT II have this particular worldview. They say the world has a status consisting of "entities" of different types; entities are characterized by the values of their "attributes"--nowadays we call these "properties"--the values of the attributes, and who are the members of the "sets"--now called "collections" or something. Status changes at points in time we call "events." Events either happen exogenously

from the outside, they are preset up before you run, or they happen endogenously. In other words, when you are in the middle of an event you can cause one or more other events to happen subsequently. In SIMSCRIPT I we had a form on which you describe the “entities, attributes and sets.” This was the Definition Form. We also had a form which was a WYSIWYG report generator form. Getting these forms out to the world was a nuisance, so we wanted to get rid of the forms. For SIMSCRIPT II, we decided to go from having Definition Forms to statements that you read—EVERY JOB HAS A PRIORITY and things like that. So that was one kind of thing we did for SIMSCRIPT II. SIMSCRIPT I was well used within RAND, so RAND got an immediate payoff. They were already building big simulations and they started working on big simulations in SIMSCRIPT. One of the things we put in SIMSCRIPT I was an ACCUMULATE statement which accumulated statistics, but it was a substitute for the assignment statement. Instead of assigning a variable, you could simultaneously assign a new value to the variable and accumulate these statistics over time. But these were all in the executable part of the code. And when we saw our first real SIMSCRIPT I programs, they were half-filled with these ACCUMULATE statements. Then it struck me that we could do the programmer a great service and substantially reduce the size of the program by just telling the program once and for all at definition time, that something was to be accumulated. We called this the “automatic accumulate statement.” Herb and Bernie and I had this big--not a fight--but a big heated discussion about whether we should put the automatic accumulate statement into SIMSCRIPT I. On the one hand, the SIMSCRIPT I compiler was done, the manual was done, we were about to put it out. But on the other hand, we could reduce the programs by half. Bernie ended the discussion by saying, “Harry, the automatic accumulate will be the first feature of SIMSCRIPT II.” It was agreed.

Yost: Was that because he wanted to generate business for CACI?

Markowitz: Bernie wasn't going with CACI. He was tired of programming.. . . .

Yost: Right. Okay. Yes.

Markowitz: You get a huge program built, you don't want to cut into it. That was it. SIMSCRIPT I was done, and that was it.

Yost: What was Herb's perspective?

Markowitz: He also felt the manual was done, that was it. Okay. Agreed. That is going to be the first feature of SIMSCRIPT II. There was a problem of how to build SIMSCRIPT II, how to make a compiler. I didn't know anything about compiler writing. I suppose it would have been nice to read a book on compiler writing or something like that, but sometimes--maybe it is a quirk of my personality--I find it is easier to figure things out by myself than find a suitable reference. Not always. Sometimes you can get some nice references. I don't really arrive at everything from scratch, but I do have a habit of going and figuring things out myself. I had heard about some language that was programmed in itself. I can't remember whether it was JOVIAL [*Jules Own Version of International Algebraic Language*] in JOVIAL. I am not sure whether that is programmed in itself, but there was some language that was programmed in itself. So that titillated me, SIMSCRIPT written in SIMSCRIPT. That meant you had to have an

entity-attribute and set description of the things around a compiler. So I sat out on a park bench at the beach near RAND for many, many afternoons trying to figure how to make a compiler. SIMSCRIPT is not an easy language to compile. SIMSCRIPT II especially. It has control phrases that you can concatenate any way you want. Like for example, you have a control phrase like, “for each machine group in shop.” You can hook on “with something true,” or “until something is true.” Then within that you can say “for each job in the queue of machine group:” then you can hook that on to a “do” statement, which will control a range of statements, or you can hook it onto a “find” or a “compute” statement, or put it on any one statement. So anyway, I figured out how to make a compiler based on an “entity, attribute, and set” view of the compiling process. We began to implement that at RAND. There was a period of time when I was working more than half time at RAND and less than half time at CACI; then a time when I was working more than half time at CACI, and less than half time at RAND. Then I decided CACI was too pressing. It was about then that IBM said, ‘We’d like you to make a new SIMSCRIPT compiler that will work on this new operating system.’ I think it was 360.

Yost: Yes, I think it was 360.

Markowitz: Yes, 360. We said, “Yes, but we are not going to do it the same way. We are not going to make it a preprocessor to FORTRAN. We’ve got this new technology for building a compiler from SIMSCRIPT into assembly language. The new SIMSCRIPT was called I.5 and it used this technology. Of course there was a question of how you, if it is programmed in itself, how do you compile it the first time? What do you use to bootstrap? Well, we got SIMSCRIPT I.5 from SIMSCRIPT I. Then SIMSCRIPT I.5 from SIMSCRIPT I.5. SIMSCRIPT II was built

on SIMSCRIPT I. Then later it was built in II. As to the history of SIMSCRIPT II--I can't remember whether I started it in 1962 or 63 or 64. The first programmer had a lot of difficulty with the thing. He was probably a very good programmer, but not that good. Bernie Hausner had been off touring the world. He even visited the Simula (*Simulation Language*) folks in Norway, and talked about SIMSCRIPT there. They wanted to go off in their own direction, which they did. Bernie Hausner toured the world, maybe for a year, and then came back. I got him back and he programmed SIMSCRIPT II for maybe a couple of years. He got it to the point where SIMSCRIPT II could compile SIMSCRIPT II. So the guts of it were there. Then he said, 'I am working with a fellow by the name of Richard Villanueva. He is as smart and energetic as I was when I first started. He'll finish the project.' And he did. In the meantime, we were looking for a writer. We got Philip Kiviat, who had a programming language of his own. GASP, I think, was his language. He came and wrote the SIMSCRIPT II manual. The three of us did the fine details of the language design until I got too busy at CACI, and then the two of them finished it up.

Yost: Was it roughly the mid-60s when things got busy at CACI, a couple years after it began in 1963?

Markowitz: Yes. Now, there are certain events where I know the dates. And then we can squeeze things between them. We know that SIMSCRIPT I was published in November of 1962. CACI was incorporated July 17 of 1962, so we had already started in business and we had all of thirty-three thousand dollars worth of revenue between the day of incorporation on June 30, 1963. Then you can see the business' build. This is just gross revenue, don't look around to see whether this is in thousands or millions, that is dollars right down to the dollar.

Yost: [Looking at the financial statement] Okay.

Markowitz: So business is picking up. Here is 1967.

Yost: Six hundred thousand in revenue?

Markowitz: Yes. Right, of revenue for the year.

Yost: What marketing was done for CACI and who was doing it?

Markowitz: The division of labor was--I was Chairman of the Board and Technical Director. I was more Technical Director. There was no Board of Directors. Herb was President, but he was the marketing and business person. We ran ads for the SIMSCRIPT courses. I don't recall whether we advertised that we would make SIMSCRIPT I.5 compilers available for other machines, but we did. When we finally sort of got off the ground, our product lines, we made SIMSCRIPT compilers for different platforms and continued to give courses. I think we had a CDC, or a couple of CDC machines, a Univac machine, another version of the IBM machine. It was a fair little department there. We used our "SIMSCRIPT in SIMSCRIPT" to bootstrap from one compiler to another. But we had to hand write the basic I/O routines and so on. Then we got a big contract. Herb had been trying to get some Washington contract. But we got a big one. We got a contract from the EDA, a part of our government that was supposed to help developing countries. We got a contract to build a database system for them. Let's go back to the grand plan

for SIMSCRIPT II because that's important here. SIMSCRIPT I was designed to be a simulation language, but as an afterthought we said, there are some useful set-processing facilities. As an afterthought we allowed the guy to put in a card that said "non-simulation," then we didn't supply a timing routine for his program. So, as an afterthought, we thought SIMSCRIPT I could also be used for other things than simulation. SIMSCRIPT II we thought of as a general programming language. Partly as an expository device, we spoke of levels of SIMSCRIPT. Ultimately there were supposed to be seven levels of SIMSCRIPT II. Level 1 was a simple teaching language. Level 2 brought you up to sort of a FORTRAN level. Level 3 got control phrases in and various other facilities. Level 4 finally told you about entities, attributes and sets. Level 5 got events and simulation capabilities. Level 6 was supposed to be a database manager. Not necessarily a database for simulation, but just a database manager. The basic idea was that SIMSCRIPT has this view that the world consists of entities, attributes, and sets. It is the world that consists of entities, attributes and sets, including the world that is represented by a database. Not just simulated worlds, but real worlds consist of entities, attributes and sets. SIMSCRIPT II level 6 was to be database. RAND implemented SIMSCRIPT II through level 5. Tell me, have you seen a...there's an Encyclopedia of Computer Sciences and that has an article on SIMSCRIPT. Have you got a copy of that? Have you seen that?

Yost: I don't have it with me. I did read it recently.

Markowitz: You have seen it.

Yost: Yes, the entry in the Ralston's Encyclopedia of Computer Science?

Markowitz: I can't remember [Marcel Dekker, Inc., New York, 1979], but if you need a copy I can supply you one. So the EDA contract was to add SIMSCRIPT database capabilities. I think we were calling it SIMSCRIPT SR, for *Storage and Retrieval*. Database was to be added to SIMSCRIPT I.5 capabilities. As a spin-off of that, we built a query facility. We called it "Quick Query." When it was internal EDA it was spelled QU, of course, *Quick Query*. EDA used that quite a bit. There was a form that you filled out which asked what kind of entity type you wanted to go fetch data from, and had selection criteria for what about it that you wanted, and so on. It had two or three forms that you had to fill out that got key-punched. We were still using key punch here. Anyway, EDA really liked Quick Query. We thought we would make a commercial product of it. We renamed it. The commercial product was QW, *Quick Query*. That was when Herb and I had our first big falling out.

Yost: This was over the pricing of this product?

Markowitz: Over the pricing of Quick Query. I wanted it to be priced cheap like under ten thousand dollars to get lots of users and then we could start jacking up the price. Our competitor was something called MARK IV. I don't remember when MARK IV....

Yost: The Informatics one?

Markowitz: Yes. They were charging twenty-five thousand or something like that. So Herb felt it would be an indignity to charge less than twenty-three thousand or something like that.

We had this disagreement. Also at that time he had hired a salesman type. He wasn't a computer salesman in particular, but a salesman type. Herb and the salesman on the one hand, and me on the other, had disagreements on the marketing policy, the pricing policy, things like that. We tried to resolve things--at one point we decided to form a Board of Directors that consisted of just internal shareholders. There were no external shareholders. But we just made a committee of the senior people at CACI and that became like a Board of Directors. Herb and I would plead our cases when we disagreed and get some kind of a tie-breaking vote from this group. Somewhere along the way Herb got tired of this. On March 15, the Ides of March of 1968, Herb, who had 47 percent of the stock, and Jim Berkson, vice president of finance, with 5 percent of the stock, fired me with 47 percent of the stock. I was fired on a Friday. On Monday I called Jack Little, who now had become a vice-president. He used to program for me at LP/I. He was now Vice-President at Planning Research Corporation. I called him up and said, "Herb fired me." Jack said, "Crazy, man. Come on over!" So I started there, Planning Research. At one point said they had this internal information system. It was in COBOL (*CO*mmon *B*usiness-*O*riented *L*anguage) and it had become very inflexible. It was hard to do anything new with. They asked if I had any suggestions. I suggested that they put in a SIMSCRIPT-like database with create, destroy, file remove, subroutines into PL/I. Time keeps advancing and now it's PL/I. We called that SIMSCRIPT PDQ, because it was put together very quickly, right. But that has a little story to it. I had been consulting two days a week, or something like that, for Planning Research for some time before this thing came up. I told them what I'd do is I would disappear from the office for two months. I would charge them for one month worth of time. I would show up two months later with

a program written by hand. I would need a couple of smart programmers to get it punched up and debugged. Then six months after that, you would have SIMSCRIPT PDQ, and it would be flexible, et cetera. In fact, it took seven or eight months to debug, not six months. The way I spent my two months was I cleaned up some affairs, I got on an airplane and went to Hawaii. I met Bernie Hausner who was in the Hawaiian airport. He said, "I understand Herb fired you." I said, "How did you hear that?" He said, "Jack Little told me." Something like that. Then I went down to Tahiti and Bora Bora. On Bora Bora, I'd have a big breakfast so I couldn't go snorkeling, so I'd program. Then I'd go snorkeling. So SIMSCRIPT PDQ came out of that. And it worked. Many years later I would hear that they were still using SIMSCRIPT PDQ and it was very flexible. But it had one problem: that if the system crashed at a particular time it was horrible to get everything back where it was. They had solved that problem. They weren't telling anybody how they solved it, but they had solved that problem. So when I got to program EAS-E, EAS-E was. . . .

Yost: This is now at IBM?

Markowitz: Yes, IBM, right. I had been warned to think about that problem. I did read Chris Date's book on--I don't know, maybe I read it after I started programming. So I have this dual career. Somehow Sparse Matrices just took care of themselves and a couple of other things I'd done took care of themselves. But I would go back and forth between finance, where there was a demand for me, and trying to get SIMSCRIPT II finally off and put together. When I got to IBM we picked up a public domain version of SIMSCRIPT II. CACI was on to SIMSCRIPT II.5

now. I took out level 5, I took out the simulation capabilities, and we put in level 6, the database capabilities. The reason I took out level 5 was I figured if I came out with an improved simulation package, CACI would sue. CACI was very litigious. While there would be no merit, IBM might just figure, well, it is too much fuss; let's just kill the project. Burt Grad, who called on your behalf, was the one who worked with me to name it EAS-E. He said, "The reason you are not getting anyplace in IBM is nobody is interested in SIMSCRIPT." You have to think of a new name. We said, 'Okay, well how do we get a new name? Well, this thing has "Entities, Atttributes, Sets and Events." Oh, easy, EAS-E. So that became EAS-E. The EAS-E software [that A Malhotrice and D. Pazel built] worked just fine. We had one big internal application. There is a big job shop within IBM Research which is called the--I can't remember--but they did all the technical projects. All the physical hardware the scientists wanted was built in this big job shop. When EAS-E was ready, we got this invitation to go build that database system and it worked just fine with multiple users and so on. But I failed to convince IBM to release EAS-E as a product. This was at a time when IBM had just converted from IMS to System R and SQL. I failed to convince them to convert to what was called a "network" database language. So at some point, I just went off and became a professor. I had done the demonstration of the thing, and it was feasible. I don't have too many unfulfilled goals in my life, but one of my unfulfilled goals was SIMSCRIPT II. As it was planned, plus some things that the world has learned since. For example, in SIMSCRIPT I and SIMSCRIPT II, we had pointer variables, but they pointed to anything. Now I would tighten them so that you could define something that would be a pointer to anything, or it could be defined as a pointer to a job, or pointer to a machine. Also, I would do inheritance like the modern object orientated languages rather than like SIMSCRIPT II's "common attributes." Those had to be located in the same place in different entity records. If we

were to redo SIMSCRIPT, if somebody wanted to make a SIMSCRIPT III, there are things which the object-oriented folks have done which we have failed to do, but would fit in. And then I think there are certain things that SIMSCRIPT does better. SIMSCRIPT was intended originally as a modeling language. You saw your world in terms of entities, attributes and sets. It is a convenient way of documenting a model, and then the programming was supposed to come easy. Now you are supposed to see your world in terms of “inheritance, polymorphisms, and encapsulation.” SIMSCRIPT people don’t see the world that way. Sets are built into the world-view. Sets are part of the modeling. Now, of course, you’ve got sets because you’ve got collection classes. But in C++ if you use one kind of collection class you’ve got to program in one manner, and if you use another kind of collection class you’ve got to program in another manner. All that is behind the scenes in SIMSCRIPT. You could define a set in SIMSCRIPT to be first in /first out, or last in/first out, or ranked by one or more attributes. That is at definition time. Then when you go to program, if you want to file something into set you say, “File job in queue.” It doesn’t make any difference how that was organized, from the modeler’s point of view. So I think there are still things which SIMSCRIPT does more neatly than object-oriented languages, but other things which object-oriented do better, like inheritance and type checking. But SIMSCRIPT could do the same if somebody wanted to.

Yost: Looking at it more broadly, was SIMSCRIPT II used extensively outside of the simulation area?

Markowitz: I think it was mostly used in simulation. It is still around. It is still being used.

Yost: SIMSCRIPT II.5?

Markowitz: It is II.5. Phil Kiviat left RAND and tried to do what Herb and I did. He had, I think he called it "SIMSCRIPT +" or something like that, but he didn't make a success of it. I was gone by that time, it was after the Ides of March in 1968, but at some point CACI picked up Phil Kiviat's version of SIMSCRIPT II, enhanced it a little more, and called it SIMSCRIPT II.5.

Yost: Did RAND ever try to hold the intellectual property?

Markowitz: No, that wasn't RAND's business. I don't mean it in any nasty sense, "that's none of their business." I mean it is not the business that they were in.

Yost: Sure.

Markowitz: The business that they were in at the time was primarily to be a big think tank for the Air Force. They did lots of things which were very closely Air Force-oriented, and they did other things which were more research-oriented. So, for example, the linear programming stuff, maybe had some direct Air Force applicability, but it was also something that was good for the world. The simulation of SIMSCRIPT I, certainly the folks in the Logistics Department, at least, used it. But nobody proposed that they make a commercial product out of it and market it. That was just not, at that time, done at all at RAND.

Yost: Was there any security issues with RAND?

Markowitz: No, SIMSCRIPT was always developed as a non-secret. In fact, when I went back to RAND, it was because my work was going to be publicly available.

Yost: Going back to CACI, was there a sense that you were part of an emerging service industry? Did you belong to ADAPSO, or did you know that ADAPSO existed?

Markowitz: I had no idea. I had no vision of myself as part of a software industry. I had a vision of us in the computer simulation business. We went to conferences where GPSS representatives would get up and say, "This is why we are good," and other people would say, "This is why we are good." So we attended those kinds of conferences. Our chief competitor was GPSS. We had never heard of Simula. I guess GASP was also being used. I can't remember. I guess there were languages used besides SIMSCRIPT and GPSS, but those were the main competitors at the time. Our position was, okay, GPSS is easier to learn than SIMSCRIPT, but SIMSCRIPT is more flexible. A lot of people started simulation by using GPSS, but would bump up against its limitations, and then they'd have to come and look at SIMSCRIPT.

Yost: Were there certain types of industries or customers for one versus the other?

Markowitz: GPSS has a model of particles flowing through stations. So if you are talking about a job shop, or a manufacturing facility, GPSS was a plausible candidate. But it doesn't have any programming facilities, or if it does, it was through a hook that really didn't help you in program. So if you tried to simulate a computer where there is some internal logic about queuing for the

CPU or queuing for I/O and so on, GPSS just doesn't give you that facility. Whereas, these are events that change, "entities, attributes, and sets;" that's just fine for SIMSCRIPT. It turned out, I understand now, that the military was a big SIMSCRIPT user. I don't know the details, but I can see that if I had to simulate a situation where two armies are attacking each other, I wouldn't know how to do it in GPSS. Whereas with SIMSCRIPT, yes, I can figure out what the entities, attributes, and sets are.

Yost: Right. Looking back at CACI, what do you see as its place in the software industry?

Markowitz: You are better able to see the industry. I can see its place in my development. I think maybe that I could give you a better answer as to my place in CACI's history, rather than CACI's place in the software industry. Of course, Herb and I started it. Okay, so you get one point for being the founder. Then we built it a certain way. We had this SIMSCRIPT business, this software development business, which now has become a relatively small part of CACI. We also had the first--mostly Herb, but with me doing technical support and then developing the product--had the beginnings of marketing to the government. We did a simulation or two, I don't remember the details, but the big thing was EDA. Then I remember one of the programmers, Ron Storts, became involved in delivering some other Washington product that Herb marketed and I helped with. So during my five years there, we got started and we made our first contacts with the Washington DC crowd. Then I left. Part of my shares were sold at the initial public offering which was already scheduled by the time I was fired. Little by little I reduced my position over the years. Then CACI had hard times where Herb, and whoever he

chose, weren't doing too well. Somehow Herb, very wisely or by a stroke of luck, got Jack London into this thing. He [London] further developed the Washington end of the business.

Yost: You said certain projects research areas of yours just took care of themselves and you were done with them. The "Portfolio Selection," have you continued to work in that area?

Markowitz: Oh, Yes. Oh, I should mention that when I was fired we were developing SIMSCRIPT Storage and Retrieval for EDA. They never were able to finish that then. It was, I think, too tough for them, I mean, the remaining crew. So I am delighted that I had the opportunity to show its feasibility with the IBM EAS-E. Portfolio Theory has become very large. I mean, Portfolio Theory is very, very well known. Of course, I got a Nobel Prize for "Portfolio Theory."

Yost: Right.

Markowitz: The year before that, I got a prize from the Operations Research Society of America and the Institute of Management Sciences, the (John) von Neumann Prize. In some ways it is more dear to me because it cited three pieces of my work: "SIMSCRIPT," "Sparse Matrices," and "Portfolio Theory." I have these four little rooms, and I make a living here. I could retire, I mean we have enough money to retire on, but I am having great fun. On Wednesday, Thursday, Friday, I work for clients in Portfolio Theory. I am also on some advisory boards and a board of directors, but that is also in Portfolio Theory. On Monday and Tuesday I do administrative stuff, trying to get my desk clean. Also if I have some time, develop a Web site. I have some software

that is set-oriented and I am going to try to develop a Web site where I am going to harangue the reader about the virtues of seeing the world in terms of “entities, attributes and sets.” I’ve laid out some money out of my own pocket to develop some things. You’ll notice on the door as you came in a sign saying “Harry Markowitz Company” and under that a sign that said--I don’t remember how much it is spelled out-- “Rational Decision Making Research Institute.” It probably says, “RDMRI.”

Yost: Yes, I noticed that.

Markowitz: RDMRI. That is a non-profit organization that’s supported by my wife and me. This way any money we’ve actually spent developing EAS-E II, or something like that, can be at least tax deductible. My only obligation is that I’ve told the IRS that I will make this available to the world, which is just fine. There’s always something else going on that seems to eat up Mondays and Tuesdays. But I am going to put together a Web site, and make EAS-E software available. But your question was, what about “Portfolio Selection.” Everybody knows about Portfolio Theory. For example, there are folks who give meetings every six months called the “Berkeley Program in Finance.” I got a call from them a few months back saying, “Our next meeting is in March of 2002. Someone just pointed out that will be the 50<sup>th</sup> Anniversary of the publication of your “Portfolio Selection.” They said “why don’t you come up and give us a little talk about the old history and we’ll celebrate your 50<sup>th</sup> anniversary.” So, it is still alive. I get invitations from all over the world to come give talks. I pick and choose places that my wife and I haven’t been to before.

Yost: Well great, and I think that concludes the areas I wanted to cover. Thank you very much for your time.

Markowitz: Good. Let me give you a quick little tour and let's go off to lunch.