An Interview with

FRIEDRICH L. BAUER

OH 128

Conducted by William Aspray

on

17 February 1987

Munich, West Germany

Charles Babbage Institute
The Center for the History of Information Processing
University of Minnesota, Minneapolis

Friedrich L. Bauer Interview
17 February 1987

Abstract

Bauer briefly reviews his early life and education in Bavaria through his years in the German army during World War II. He discusses his education in mathematics and theoretical physics at the University of Munich through the completion of his Ph.D. in 1952. He explains how he first came in contact with work on modern computers through a seminar in graduate school and how he and Klaus Samelson were led to join the PERM group in 1952. Work on the hardware design and on compilers is mentioned. Bauer then discusses the origins and design of the logic computer STANISLAUS, and his role in its development. The next section of the interview describes the European side of the development of ALGOL, including his work and that of Rutishauser, Samelson, and Bottenbrach. The interview concludes with a brief discussion of Bauer's work in numerical analysis in the 1950s and 1960s and his subsequent investigations of programming methodology.

FRIEDRICH L. BAUER INTERVIEW

DATE: 17 February 1987 INTERVIEWER: William Aspray

LOCATION: Deutsches Museum, Munich, West Germany

ASPRAY: This is an interview on the 17th of February, 1987 at the Deutsches Museum in Munich. The interviewer is William Aspray of the Charles Babbage Institute. The interview subject is Dr. F. L. Bauer. Let's begin by having you discuss very briefly your early career in education, say up until World War II.

BAUER: I was born in the year 1924 in a town in Bavaria, the name of which is Regensburg. For the first eight years of my life, I was in a small village near Regensburg. I went to elementary school there and it was the most interesting sort of elementary school. It was one classroom with pupils of all ages. It was very interesting when the teacher told stories about history to the "grown-up people", while the small ones were allowed to do some exercises in writing letters or in arithmetic. I was always greatly entertained at that school. Later, when I was eight years old, my parents moved to Pfarrkirchen, a small town in Lower Bavaria and I continued elementary school there. Now I had a different situation, being no longer in a rural area, but in a small town; and the classroom work was much less interesting. From 1934 to 1939, I went to secondary school there, and then I had to move to Munich to continue secondary school, from 1939 to 1942. In the year 1942, I finished secondary school. In the meantime, there was war -- practically all the time I was in Munich, there was already war. Till 1942, I was not much bothered by the bad effects of war. Of course it changed then. I was drafted to the Army and spent the next three years with the Army in the infantry, most of the time in Russia. Early in 1945, when I was on the western front, I came with gunshots to an American prisoner hospital (where I received by mistake the Purple Heart), and then to a prisoner camp near Cherbourg.

ASPRAY: Just to get it on the record, there was nothing during your wartime experience that contributed to your scientific education?

BAUER: No, emphatically nothing except for a funny little incident during my time in Russia. I was given a little booklet, a Goeschen booklet on algebra written by Hasse. Unfortunately, it was a volume on exercises to the Hasse algebra, and I didn't have the Hasse book on algebra. When I tried to reconstruct the book from the exercises, I was only partly successful in doing so.

Practically, there was at that time no opportunity for me to do mathematics and I did not have the good luck to avoid the draft.

ASPRAY: Yes.

BAUER: After coming back home in September 1945, I worked in bookkeeping for my father. Early in 1946, the University of Munich opened again. I applied and I was admitted to the University of Munich to study natural sciences. The field I was most interested in was mathematics, together with physics, astronomy, and logic. I considered mathematics more or less to be my major subject but I did not neglect physics and astronomy. Logic was kind of a peripheral field because they offered lectures in logic, but their offerings were somewhat restricted.

ASPRAY: Yes. Can you mention some of the instructors?

BAUER: Yes. I had the luck that some of the retired professors did come back to the lecture hall. There was Oscar Perron in mathematics, and Heinrich Tietze and Konstantin Caratheodory. On the physics side, Arnold Sommerfeld. It was a great impression these gentlemen gave me in these early times of my scientific upbringing.

ASPRAY: Yes.

BAUER: Maybe I should mention that mostly I was under the impression of Perron. It is reflected in the fact that I considered myself, for the rest of my life, to be basically an algebraist. But in 1947, that is in the middle of my studies, I was very much attracted suddenly by theoretical physics, particularly by a modern part of theoretical physics, by elementary particle theory. I then continued to combine mathematics and physics, or better, to combine algebra and theoretical physics, algebra and elementary particle theory. I did so under the direction of Professor Fritz Bopp, who was the post-war successor of Sommerfeld. For a while, it looked like I would be sidetracked to theoretical physics. But by some strange coincidences, very soon I had information about the efforts on computers in the United States. I was interested. I looked for publications. I looked for books, and I had friends that gave me this article or that note on what was happening. I was also asked by Professor Bopp to supervise the mechanical laboratory in building a small analog computer, a kind of a Bush-type analog computer, which I didn't like very much. I tried to convince the professor that it would much better to build it by electric servomotors. But the mechanical solution seemed to be simpler to him. So I sort of had the reputation that I was somehow familiar with computing.

ASPRAY: I see. What sorts of things did you know about from the American scene at this time?

BAUER: Mainly, the normal newspaper information, a few specific articles published, but not very much. But by this reputation that I was interested or that I was somehow doing something about computation, I was then coming into contact with a group that was just being formed at the Technical University of Munich under the Professors Hans Piloty and Robert Sauer. They started a clandestine seminar on computers. By a friend of mine, Hermann Jordan, who happened to be an assistant to Professor Sauer, it was managed that I was allowed to attend that clandestine seminar with the provision that I should keep my mouth shut. And so at that seminar, we for example, discussed the von Neumann report which Sauer had managed to have access to. The report was in the open literature at that time, but it still was not so easy to have a copy of it. The von Neumann report probably was the first detailed information I had, and also the publication by Stiefel, Speiser, and Rutishauser, which came out a little bit later. Those were the first two solid bits of information that I had.

ASPRAY: When you say the von Neumann report, do you mean the 1945 EDVAC report or the first Institute report?

BAUER: The first Institute report.

ASPRAY: Goldstine, Burks, von Neumann?

BAUER: Yes. That is the one that I have in mind. One of the particular things that brought me in this line was also that I was, by one of these strange coincidences, pretty early informed about Shannon's work, and also about Hamming's ideas. And so, coding and practical applications of coding were, at that time, in the line of my interest. I took it up very seriously and sometime in 1951, I even submitted a patent application on a particular error-correcting code. That came out more or less from the stimulation I had from Shannon's work and from Hamming's papers. But most of my interest in computers was, of course, raised by the Piloty-Sauer seminar. The reason for this seminar became very soon clear to me. These two professors had a plan to build an electronic computer in their institute, in the Technical University of Munich. So this was a preparation for this undertaking. But I was still in theoretical physics, sort of speaking, and this went on. I got my Ph.D. in January 1952 with a thesis on elementary particles and with a strong inclination to group representation theory. In the same year, practically one month later, I also had my first meeting with Rutishauser in Zurich, whom I could visit on one of the very first opportunities I had to leave the country. At that time a German could not leave the country unless he had a permit from the military government. And so, meeting Rutishauser and seeing in operation Zuse's Z4 -- that was another refreshing impulse. By the way, I should mention that I had met Zuse already quite early. In 1947, my professor in logic, Britzlmayr, had invited Zuse to a seminar talk, and Britzlmayr was interested in one particular application to logic Zuse had published, an application of his Plankalkul. As a very simple example for his Plankalkul, Zuse had written an algorithm for testing whether a propositional formula is well-formed. It was a problem which was at that time sort of in the air; but was not yet everywhere fully discussed. So Britzlmayr was interested. Zuse did come, but did speak more about the Plankalkul than about the particular example in propositional logic. He also spoke about his computer. It was a great confusion in the seminar because most people really didn't understand anything. They had expected something very different. Anyhow, I was at that early moment informed about the Plankalkul; but I also had my difficulties to see all of it, particularly since Zuse's explanations at that time were not always the most clear ones. But it certainly, subconsciously, has been one of the stimulations that influenced me. Also, coding theory was something that I became interested in, partly because I had realized that all these things could be done by binary logic -- this was clearly said in Zuse's Plankalkul. It sort of supplemented from the side of a practical engineer what was shown in theory by Shannon about the possibility of switching circuits.

ASPRAY: Did you have access at this time to the Bell Technical Journal?

BAUER: If you have in mind the Shannon paper?

ASPRAY: Shannon's papers...

BAUER: In these years, I did run into the papers. I remember this quite well, since I studied them carefully. For example, the one on coding theory and the one on secrecy systems. I was aware of them.

In 1952, a number of events sort of supported my subconscious decision to leave theoretical physics. The decisive step then came in the middle of 1952 when my friend Hermann Jordan one day said, "I'm going to a position at another university." He was assistant to Sauer, who was a mathematician and he said, "I go to a physics professor," because he was a physicist in the sense he understood himself. So he said, "Sauer then has an open assistant position, and I will recommend you to him." I said, "Fine," and to my great surprise, Sauer, who already knew me from the seminar, accepted me. And so in the fall of 1952, I then went to Sauer. I left theoretical Physics. I left Professor Bopp. I went to a completely new field and disconnected myself from elementary particles. My work with Sauer, of course, was at first the normal work of an assistant to a professor. I had to do the grading of papers, and the supervising of theses and all these things. But, also I helped in building up the group on the computer, which was then termed PERM. Pretty soon I found myself in the middle of the PERM group. One of the great things was that I could also bring one of my very, very close friends, Klaus Samelson, into this group and so we were working together from 1952, practically for the rest of Samelson's life.

ASPRAY: What were your responsibilities in the PERM development?

BAUER: Strictly speaking, purely on the programming side, since we were not supposed to understand anything about circuitry. But, of course, we couldn't help; we sometimes understood a little bit about it. Then we started telling the engineers that they could do something better, which the engineers didn't like. In the first line we had to prepare ourselves to program the computer. We had to develop what one called at that time a program library. We prepared subroutines for elementary functions, then for common algorithms, and so on and so on. We also prepared ourselves for the programming task and found out by studying, for example, the book by Wilkes, Wheeler, and Gill that we had to do a lot about testing and about post-mortem dumps and all these kinds of things. This led us, more or less directly into algebraic compilers, because the dirty work of machine coding already and at that time, seemed to be replaced better by some better techniques. We also influenced a little bit the design of the PERM on the functional side. We started investigating floating point. First, the computer was planned to be just a fixed-point computer, like the computers used to be at that time. We convinced Piloty that it would be important for the difficult computations that were planned to have floating points. So the computer got wired-in floating point arithmetic. It was one of the very first computers with floating point arithmetic. It was also a parallel computer. That means, being parallel and having a wired-in floating point, it was quite quick in floating-point computations. There were a lot of problems, of course, connected at that time with developing sound floating-point arithmetic.

Another thing we influenced was the addressing system. In our group, following a proposal by our co-worker Heinz Schecher, indirect addressing was invented. We developed all kinds of sub-routine linkages, mainly with the intention that such a fast computer should not be slowed down by a lot of organizational overhead.

So work went on. I mentioned before compilation problems. We soon found ourselves on the way to algebraic compilation.

ASPRAY: When did this work on algebraic compilation begin, would you say?

BAUER: I think we started immediately after the more basic problems like the elementary functions and the floating point arithmetic were under control. At the end of 1954 or the beginning of 1955, we addressed the problems of algebraic compilation. The first published reference to it is from November, 1955 at a Dresden meeting where Samelson gave a lecture on the problems of compiling algebraic formulas.

ASPRAY: Were you familiar with the work of Laning and Zierler at this time?

BAUER: No, not at this time. In fact, certain roots of our approach probably had been earlier than Laning and Zierler's paper. I should come back to the professor in formal logic I mentioned before, Professor Britzlmayr. You remember that I said that Zuse gave a seminar talk on how to test a formula to determine whether it's a well-formed formula of propositional calculus. Roughly at that time, a co-student of mine (the name of which is Helmut Angstl) came to Professor Britzlmayr with a simple mechanical device that would also do the test. It was made from wood, and it would move some lever and some bars would fall down. If the formula was well-formed, then they would fall into the right gaps, and the whole thing would be okay. Britzlmayr said to me, "Do you think such a thing could be used?" I said, "Yes, of course, but not mechanically; this has to be done electrically." He said, "Could you do the wiring for this?" I said, "Of course". So I made a wiring design for it at the turn of the year 1950 to 1951. At that time, Polish notation was fashionable among mathematicians. So I accepted that the input will be in Polish notation, as it was, in fact, considered by my co-student, Angstl. If the input is in Polish notation, the problem is quite simple.

ASPRAY: Yes, it is easier to parse.

BAUER: Easier to parse, yes. So I made a wiring design for it and showed it to Professor Britzlmayr who said, "We should build it!" I said, "Yes, I shall build it." But I didn't immediately have time for building it.

The whole idea of this was, of course, the root for our approach to algebraic compilation. We knew how to parse mechanically a formula written in Polish or in reverse Polish notation. So when we discussed in 1954 or 1955, Samelson and myself, how to do algebraic compilations, the whole question was: now we have parentheses, and what we do now with parentheses? The solution seemed to be obvious one day; I cannot say whether Samelson or myself... One of us said to the other one, "It's quite simple. We have to push down the parentheses too; because before we had pushed down our intermediate results for Polish notation, so we have to push down the parentheses too." That meant that we had an on-line method for transforming algebraic notation into one-address code or three-address code.

Now we were aware at that time of Rutishauser's paper from about 1951 on formula translation. Rutishauser had a different method. He had a method that would walk back and forth and would work down from the top of what he called the parenthesis mountain. Our method seemed to be much simpler; in particular, it didn't walk back and forth -- it was monotonously running. It went proportional to n if n is the length of the formula, and not to n^2 -- which happened with Rutishauser's method. So we considered it to be much simpler and much more efficient.

That started the whole thing on our side of the ALGOL program language --because very soon we found out that not only our arithmetic formula could be parsed this way, but practically everything that you would want to write down in a program, provided it has nested structure.

ASPRAY: Yes.

BAUER: So, our approach to algebraic compilation had its first start with this logic computer -- we gave it the name, STANISLAUS, and it still exists. It will be on exhibit in the Deutsches Museum, I hope, next year. This first tiny thing, then the application to general algebraic formulas, and then the extension to arbitrary nested programming structures.

Our approach finally led, on the European side, to ALGOL. In 1957 we had a fully developed algebraic language -- one of the predecessors for ALGOL 58 we had worked out in a working group, where Samelson and myself were from the Munich side, Rutishauser from Zurich, and Bottenbruch from Darmstadt. The four of us approached the American side to ask whether they would be interested in this kind of thing. It turned out that there had been similar interests in doing something against proliferation of compiling languages and they wanted, at that time, a uniform or universal language. So this lead to the ALGOL 58 conference, which was held in Zurich.

ASPRAY: Would you please give me some information about the ALGOL development from the European side?

BAUER: The ALGOL development on the European side started when Samelson and myself, having contact with Rutishauser (who had done his own work on what he called "formula translation"), when we said we should aim at some common basis of cooperation; at the time it just meant that we could exchange programs between Munich and Zurich. Bottenbruch at Darmstadt, another place where computer development happened in Germany, also showed interest; in fact his interest was also quite on the theoretical side. So, we said, we could make us all a group where we would aim at the possibility of exchanging programs by using a common programming language. The general line was clear from the pre-history: Samelson and I, we would be sort of the task force on the parsing side. Rutishauser, who had already concentrated on a so-called programming system for his machine, ERMETH, that meant who had a kind of a running system that was to be used for subroutine linkages and similar organizational things -- Rutishauser would supply his particular experience on this side where he was more experienced than we had been. Bottenbruch was strongest, I would say, on purely theoretical matters, the connections to logic. He probably was also the man who coined at that time the word "algorithmic language" (algorithmische Sprache) at least in Germany.

So we started a kind of a cooperation and it developed into a preliminary report. We gave the report the final shape in a meeting we had in fall of 1957, where we all met in Lugano in Switzerland and combined a week of vacation with work. With this report as our position paper, we then approached both some American colleagues and some British colleagues. I had contact with people like John Carr, Saul Gorn, Alan Perlis in the United States, and a few more. Their reaction, very promptly, was, "Yes, we're interested and we should try to come to some mutual agreement." We also approached some people in Great Britain (Maurice Wilkes for example), but there was no reaction at that time from Great Britain. So, the English were simply left out at that time because of not showing interest.

The whole thing went on quite quickly. In a very short time we had some exchange of our preliminary papers. We then agreed on a meeting that we had in Zurich in May of 1958, which in about 8 days hammered out the first ALGOL report, the ALGOL 58 report. At that time, the American side called it IAL, the International Algebraic Language. We called it ALGOL. We agreed on everything except the name. Later, the name ALGOL was at least kept in Europe. In the United States, I think, the IAL is still existing in words like JOVIAL, ending with IAL.
That was 1958. From then on, the ALGOL work is very well recorded in the literature. A great interest came up in

Europe. It was almost a wave of enthusiasm, and many other people joined it. There were people in the Netherlands, like van Wijngaarden, and the young Dijkstra. There was Naur in Denmark, there was Woodger in England; and there were French colleagues. So, on the European side, the interest was running very high. We sensed a strong, driving force from the European side. We then went into the ALGOL 60 development. The meeting on the ALGOL 60 report was in January, 1960 in Paris. From then on the history, the rise and decline of ALGOL is well known.

ASPRAY: Yes. Would you like to make a couple of remarks about your own personal attitudes why ALGOL did not catch on so well in the United States?

BAUER: That's very difficult. There could be a number of reasons; and it's difficult for me to find out which ones are the responsible ones and which ones didn't really mean much. There is quite definitely a difference in attitude between the European side and the American side with respect to ALGOL. The Europeans thought it something they had developed, so that gave the European side a great impact. Normally, progress came in one direction, from the United States. But they had been involved -- that gave the Europeans quite a strong inclination for ALGOL, maybe even they were attached to it. Nothing of this sort happened in the United States. The United States people saw just one among many possible approaches. Even the American friends that worked in the committee probably showed less enthusiasm about ALGOL 60 than the European people did. Understandably.

ASPRAY: I see.

BAUER: But I don't think that this was responsible for the things that happened later. If some circumstances around IBM had been a little bit different, ALGOL could have developed probably in quite a different way. But it didn't. In fact, in the United States it didn't really survive.

ASPRAY: All right. Let's move on then to what happens next in your own personal career.

BAUER: Oh, my personal career. Well, apart from these programming things, of course, I considered myself in the

1950s and all of the 1960s to work in numerical analysis. I had most of my publications in numerical analysis. Mainly, eigenvalue problems and rounding error analysis.

ASPRAY: Is this a subject that had been taken up by a number of researchers in Europe at the time, or were you uncommon among your colleagues?

BAUER: The man who worked very closely together with me was Rutishauser again. I had a number of publications coauthored with him and we very often supplemented each other in this field. Apart from Rutishauser, I didn't have so many similar coworkers in Europe or in Germany.

In the late 1960s, I got more and more interested again in programming and I went really into computing, and I left numerical analysis behind. I even changed from a chair of mathematics to a chair of computer science in the early 1970s at my university. Till today, I am working mostly in programming theory and programming practice. My numerical analysis work is past history now.

ASPRAY: I see. Can you make any comments about the contribution, can you give me a summary of the contributions you made to numerical analysis? Is there something that you'd like to say about this?

BAUER: It's difficult to say in a few words, particularly since many technicalities are involved. But I think my work was always concentrated on algorithms and, in this case, more on linear algebra algorithms and not so much on differential equations. Eigenvalues, I think, was the central point of my algorithmic attitude; and continued fractions, an obsession that I had in common with Rutishauser.

ASPRAY: Was your work driven by practical applications that had arisen?

BAUER: Yes. There was one particular practical application that had shown me the way to do the algebraic problems. It again means coming back to my situation when I was assistant to Professor Sauer. Sauer's colleague, Hans Piloty, the electrical engineer who was responsible for the hardware side of the design -- his particular field was electrical filters. So he had to calculate the roots of polynomials to about 30 figures. I did not like it very much to have to compute to 30 figures, and I tried to find out stable ways of computing. All that was wanted was a Hurwitz decomposition of the polynomial, but in order to do so, nothing better was known at that time that to compute all the roots and then to compose the wanted polynomial from the roots in the upper half plane. So I tried to do direct factorization of a polynomial. That's what brought me into contact with problems of numerical stability, with problems of rounding error behavior, with all the algebraic problems with algebraic roots, and of course with eigenvalues. So, more or less from this experience did my orientation to algebraic numerical problems come.

ASPRAY: Okay. Why don't we turn now to your later career and contributions to programming methodology and programming theory.

BAUER: I think this is not yet history, and I'm still in the middle of problems of programming. What I do at the moment I would consider to be programming methodology. The particular line I'm following now since 1974 is what we call Program Transformation. We start with a program and try to develop from that program a more efficient program, or a more lucid program, or a program that in any other way is to be derived from the given program. Now, of course, that means that the given program is not necessarily a program in the traditional sense. It is not necessarily an imperative program. It could be a functional description. It could even be completely descriptive without any algorithmic content. So today it means we start from a problem specification and we study how we can derive operational programs from problem specifications. I think this problem in the widest sense includes all the practical parts of traditional programming where once you come down to operational programs you then can go through all kinds of programming styles, and you may go down to the machine forms of programming. The most interesting part of it, where I still see a lot of fascinating things to be done, is on the high side of the specification. We really start today with a specification written in predicate logic, normally first-order predicate logic, and try to distill from it not only one program, but a succession of programs of increasing efficiency. Is there a way at all to approach in general such a problem, and if so, which efficiency measures would you like to have and what other characteristics of the derived programs -- it's a fascinating subject to my taste; but I'm in the middle of it, so it's certainly too early to say something about it.

ASPRAY: It's hard to get perspective, yes. Well, you can tell me at least, how you got involved in these questions in the first place.

BAUER: Yes, that may be part of the history. After I mentioned ALGOL 60, I didn't mention ALGOL 68. I probably didn't mention it because it was the greatest disappointment in my scientific life. Several things had happened around ALGOL 68. That the work of the committee was done in a very inefficient and sometimes even very nasty way -- that had always been a very discouraging thing to me. Apart from all these circumstances, the outcome of ALGOL 68 was not what I had hoped for, or what I had wanted.

I started then, later on, to analyze what was wrong with our request; why did we fail in ALGOL 68 when ALGOL 60 was taken, at least by the scientific community, in the United States? Well, ALGOL 68 wasn't accepted by the scientific community; in fact, a number of very knowledgeable people expressed very clearly their disappointment about ALGOL 68; Dijkstra, Hoare, and so on. So I tried to find out, accepting the fact that they had reasons to be disappointed about certain aspects of ALGOL 68, I tried to find out...

TAPE 1/SIDE 2

BAUER: ...what could have been the reasons for the non-acceptance of ALGOL 68. I thought one of the things that was lacking, was a common understanding of semantics; so one should have a better formal semantics. But, then the question was: is it possible that there is a different semantics to every programming language? I didn't believe this. I said, there can only be one semantics and we can give it different forms in different programming languages; but I do not believe that there can be 15 different semantics -- there shouldn't be Russian programming and Danish programming and Italian programming, or programming on the Mississippi. So what is the common basis of all the programming languages, was my question. And then, find for this common basis a solid semantics, and try to find out what you can do with it; that means how many different programming languages you can formulate on that basis by favoring this style or giving preference to that style. Style, not to speak of notation. Notation already is different from place to place. But even style can be different, and orientation can be different -- we have more imperative languages, we have more functional ones, we have APL sorts of languages, we have LISP sort of languages -- they all should have a common basis. Once we found out the common basis, we can derive different programs from the same specification, as I explained before. These different programs could mean different programs in the sense even of different programming languages. So the bad experience with ALGOL 68, more or less, brought me in 1973 to start the whole line of work that I'm still in the middle of. There is rarely a bad thing in the world that would not have an advantage.

END OF INTERVIEW