

botMy REXX Compiler Era, April 1985 – August 1993

By **Willi Plöchl** (originally a chapter of his “Erinnerungen” in German, translated by **Walter Pacht** into English)

In this chapter I will become euphoric and effusive and will often praise myself. This is no wonder since this time was my best and most productive at IBM and I remember it gladly.

In April 1985 I was given the task to investigate if it were possible to build a compiler for the programming language REXX¹. I had no idea or a rather vague one what REXX was. Well, REXX started out as a replacement for the clumsy EXEC languages² that were used in VM³. Over time it became a full-fledged programming language. Its creator, Mike COWLISHAW, had the splendid idea to replace those awkward EXEC languages with a real elegant one, and he accomplished that all by himself. Not only did he design the language but implemented at the same time an interpreter for it. REXX evolved with an ongoing exchange of suggestions and criticisms between COWLISHAW and REXX users. Eventually it became so popular in the VM community that it was officially adopted by IBM and the clumsy EXEC languages were no longer used. Manuals (very good ones) were written and when I started to get involved with REXX it was already a full-fledged, extremely usable programming language, not just a replacement for the old EXEC languages.

When I started with REXX I didn't know all that. Neither was I interested in "playing around" with VM. I used it just as much as it was necessary for my work. I wasn't, and still am not a "hacker" who enjoys snooping for the internals of a computer system. I rather developed and built such systems. When my better informed colleagues told me that REXX were a replacement for the EXEC languages, I was skeptical in the first place. But I studied courageously the pertinent manuals and changed from Saulus to Paulus. What an enjoyable, powerful, and yet easy to use programming language! It could be used for all sorts of things and could be acquired much easier than COBOL or PL/I or C or ALGOL or PASCAL or ... I became enthusiastic and studied the manuals carefully. I wrote – as an exercise – smaller and larger programs. I participated actively in miscellaneous REXX fora in VNET⁴ and very soon I knew REXX very well. After this study period I went to Paul OLIVA (who was then the manager of the one person team Willi Plöchl) and told him what I thought of REXX. He was satisfied since he had expected this positive result.

¹ Restructured eXtended eXecutor

² "EXEC-Language": A "command language" for controlling the VM operating system

³ "Virtual Machine": A System that lets the user believe that she or he is working on a real computer. In widespread use in IBM and at large customers.

⁴ VNET: An IBM-internal system, similar to today's INTERNET.

Of course I was aware of a genuine disadvantage of the REXX system: It was rather slow since the programs were interpreted⁵. Interpreting had, however, the tremendous advantage that testing programs was very simple and that the effect of program changes could be immediately observed.

Many IBM customers used REXX already for application programs and they should be more efficient. The desired run time improvements could only be achieved by compiling the programs, i.e., translating them into machine language that would be run at high speed. And such a REXX compiler didn't exist then.

Therefore I started to analyze the REXX Language's peculiarities with respect to the possibilities of compiling it. I realized very soon that there would be problems since an interpreter's dynamics cannot easily be reconstructed by a compiler. In particular, REXX has extremely dynamic constructs such as 'INTERPRET' which allows to interpret statements that are generated at run time. I concluded that many functions were to be implemented by invoking code in a run time library. I kept taking notes in my 'work book' which I kept in VM and printed only occasionally when needed. Therein I collected my ideas regarding possible solutions and reservations. I devised instruction sequences, designed a compiler structure and pursued the project with utter dedication. It pleased me more and more and what was very pleasant indeed: nobody interfered and I could do what I wanted. Occasionally I reported to WALK and OLIVA but neither bothered me in any way. I could live out my phantasies. Wonderful! I requested and got the interpreter's program listings in order to study how COWLISHAW processed the program instructions. In all these activities I benefited from my rich compiler experiences. I knew what had to be done and didn't bustle around without a plan. It was a marvelous time.

Someday someone brought the news that the "Haifa IBM Scientific Center" (= HSC) in Israel worked on an experimental REXX compiler. This caught my interest and I contacted the involved persons (unfortunately I remember only three names: Pnina VORTMAN, Igal GOLAN, and Avi TEPERMAN). I asked them to send me as much pertinent information as possible, which they did. Among other documents there was an "interim report" which very well listed the problems in compiling REXX programs. Some of these I knew already, others I learned from that report. One recommendation was not to generate machine code but so called "threaded code" which is a string of pseudo operations that have to be interpreted at run time. I listened to these words and took them to my heart. OLIVA, my energetic manager, was also in touch with the HSC and at some point in time he learned that an

⁵ They were not translated into machine language.

international REXX meeting was planned where Mike COWLISHAW would participate. He asked me if I would be interested in attending and, of course, I was. There I would meet the people from whom I could learn a lot. OLI arranged the trip (he liked to do that) and we flew to Israel (the first time ever). I don't want to give a trip report. Only so much that we went with a rented car from Tel Aviv via Jerusalem through West Jordan to the Golan Heights and on to Haifa. The HSC was situated on the area of the "Technion", a very large campus on top of Mount Carmel. Our first and lasting impression was that the Technion students worked seriously and diligently. No propaganda postings or concert announcements could be seen on the walls (have a look at the university building in Vienna's first district). Students didn't wear dirty or sloppy cloths and neither lingered they on the floor or on stairs. I started to understand how the small state of Israel could withstand the pressure of its large Arabic neighbor countries.

As the name implies, the HSC was a scientific institute that did not develop programs or machines to be sold to customers. They investigated basic problems and it were scientists who worked here. The meeting, attended by Israelis, American colleagues, and COWLISHAW, the father of REXX, turned out to be very interesting and informative. Thanks to my compiler experience I could actively participate and get close to the Israelis (in particular Pnina VORTMAN) and Mike COWLISHAW which turned out to be later on very valuable. The end result of this meeting, which lasted several days, was that a REXX compiler would be desirable – BUT and then came the difficulties already documented in the interim report. By the way, the Israeli colleagues told us that they were working on a prototype for a small REXX compiler. This got us curious and we returned with a wealth of information to Vienna. Further studying and then the news from Haifa: the prototype works. I traveled again to Haifa, this time all by myself. And indeed, simple programs could be compiled and the threaded code could be interpreted. The 'proof of concept' was done (nobody had doubted that anyway). Once again I molested the colleagues with questions of all kinds and returned wiser than before. Work was underway, but it wasn't an official project but more or less a study group's effort.

It was high time to "legalize" the work we invested. To this end the Laboratory management, with the help of us professionals⁶, put together an 'initial business proposal'. I was sent to Endicott, NY, in order to present the project and to get the approval from the American management. First I presented our plans to a gremium of 11 professionals including Mike COWLISHAW. They were impressed. In particular they liked my idea to use what I called 'super digits'⁷ for the arithmetic operations. The meetings discussed not only the REXX compiler but also other REXX language

⁶ qualified technical or scientific employee.

⁷ "Super-Digit": A 32-Bit-Word of /370 is used for one digit. This way arithmetics can be performed with arbitrary large precision.

aspects. All that was very interesting and I had no problem at all but at the end of my stay I had to discuss the financial and organizational aspects of the project with the responsible managers. Norbert TEUFELHART, our expert in the Vienna lab, had briefed me very well but I was afraid of that meeting. Here I could, when acting clumsily, render down my technical success. But everything went well, management was sympathetic and they promised to approve of the project and to provide the financial support. This took a load off my mind and joyfully I traveled back to Vienna..

I continued development of more and more details and looked at DBCS⁸ support for, e.g., Japanese glyphs. I wrote a REXX tokenizer based on the concepts of a “state machine” and proved thereby that this elegant method could successfully be used to tokenize REXX programs.

But I needed urgent help since the workload became too much for me. I asked OLI for support. He suggested that I name some colleague in the lab and he would try to set him or her free for me. My choice was Engelbert MOSER, an old colleague of mine when working for the RPG compiler. OLI managed to get him into the REXX group and there was young programmer, Klaus HANSJAKOB, who worked in another project and wasn't happy with his manager (a mutual feeling). He asked me to request him if I needed support for my REXX project. This is what I had intended and sometime later I was successful. Engelbert and I agreed to split the tasks: He should take care of the runtime routines and I should concentrate on the compiler proper because of my experience in that field. I had the idea to visit with customers who used REXX as programming language for application programming to explore their experiences and requirements. Engelbert and myself put together a questionnaire and visited customers who were very pleased with our interest. They answered readily our questions and gave us insight that we could use in our future product. So we learned which areas of REXX were important for improvements and which were hardly used and therefore didn't need optimization. We maintained this customer feedback up to the end of the project. This was one reason for the unbelievable popularity of our product.

Avi TEPERMAN who was at the Endicott meeting mentioned above suggested that I should go to Haifa once more in order to discuss their experiences with the prototype. I asked Engelbert to join me on that trip so that he would become acquainted with the Israelis. As if he had known that he would become the manager of the REXX compiler project he took the initiative at the discussions and I gladly left that to him. I sort of sensed what lay ahead. Back in Vienna we continued to work applying the insights gained in Haifa.

⁸ DBCS: Double Byte Character Set (2 Bytes for 1 character).

Soon we got our next coworker, Klaus HANSJAKOB whose manager was glad to get rid of him and we were happy to get an extremely intelligent and motivated colleague.

He quickly learned the ropes and I realized what a jewel I had asked for: His previous manager must have been a dumb-head by not recognizing and facilitating Klaus' talents. Now we were three "professionals" and we got along splendidly.

Someday Fred SEMTURS (who became OLI's superior) contacted me and invited me to a confidential talk in the Chinese restaurant "Asia" in the first district. After a silent pause he asked me somewhat ironically whom I would suggest as a manager (maybe he thought I would choose myself). "MOSER" I answered. "Well" he said, "MOSER becomes REXX compiler manager. But you will be responsible if the project fails." "OK" I replied, since I was convinced that Engelbert would be an excellent manager. A few days later Engelbert's promotion was announced.

This was definitely one of the best decisions during my IBM career. I had chosen the best manager I ever had. I don't know whether SEMTURS told Engelbert about the circumstances that led to his promotion. I told him on the occasion of his 50th birthday.

MOSER turned out to be a gem. He was a great organizer, technically competent. He didn't interfere with the professionals, he took nuisances and distractions off their shoulders, defended their decisions against others, etc. etc. He accomplished to weld the REXX people together to the technically best and most motivated team of the laboratory. Their team spirit and determination for performance were outstanding. Soon enough we became the showpiece and model team for the entire lab. After this self-praise back to the REXX compiler history. Our project was officially approved. In order to absorb really all experiences of the HSC we invited some of its people for a week to Vienna and kept one of them for an entire month. But we needed more workers and succeeded to acquire Alfred GSCHWEND and an American VM expert, the young Mark TURNER for the team. Later we were joined by Klaus ZIMMERMANN and the pretty and brainy Regina PUCHER. All that was still too small a team for the compiler development. Most of the colleagues were needed for the development of the runtime routines. (There were enough people with compiler experience in the lab but they were preoccupied with other projects.)

We decided to entrust a subcontractor with the compiler development with me becoming the technical leader. We invited a few software companies to talk with them about our goals and to examine their competence. Our partner of choice was

finally the company **mbp** (= **m**athematischer **b**eratungs- und **p**rogrammierungsdiens**t**) from Dortmund. The persons who were to become our partners (N. CHRISTENSEN, Lothar SIEBERT, Oana KIPPS, J. ROBERTS, Roger SPENCER and P. KÖRBER) left a very competent impression and, as it turned out later, they really were.

We sat together to discuss the project and I presented my suggested compiler architecture. This was in principle accepted: a "state of the art"-compiler that was to generate **machine code**. mbp suggested a special way of optimizing DO-loops, IF- and SELECT-constructs (in particular with respect to compound variables). We accepted that proposal because it promised a significant improvement of the compiled program's execution speed. I specified the code that had to be generated in the so called "Mapping Document". Over time this became the compiler peoples' bible, the directive which had to be followed to the point. Now the main problem came up: How should mbp implement this compiler? mbp had no VM. After lengthy discussions we decided to write the compiler in the C programming language⁹ to test it under MS-DOS and to port it later to System/370. Furthermore we decided to rigorously apply the proven system of inspections in order to produce a quality product. As the technical leader I had to be prepared for frequent trips to Dortmund, following the development progress and keeping the mbp people abreast. We agreed on this mode of operation and started the work vigorously. Being the owner of the mapping document I worked closely with Klaus HANSJAKOB. This cooperation was both beneficial and enjoyable. We helped each other wherever possible, compromised where necessary and learned from each other. When either of us had a better solution than the other it was readily accepted. We wrote our IPFS (Initial Programming Functional Specifications) which were quickly approved. mbp prepared a high level design of the compiler and presented it in Vienna. Everything looked well until ROBERTS explained the "backend". I couldn't believe my eyes and ears: He had chosen the "threaded-code" concept that we had discarded at our earlier meeting in Vienna when we decided to generate machine code. I was indignant and so were all Viennese REXX developers. We had quite a row because ROBERTS was convinced that his concept was superior, behaved like a missionary, and didn't accept any good arguments. The situation was quite alarming. Lastly we offered two choices to mbp: either to redesign the backend or to lose the (quite lucrative) contract with IBM. ROBERTS seethed but our threat was effective. mbp promised grudgingly to deliver a new design within a month's time. ROBERTS was very angry with me (for weeks or even months) but he did his work (and very well at that). After he had resigned to the fact that he couldn't realize *his* concept he became very cooperative. When these initial hurdles were overcome the cooperation with mbp was very effective. I had, however, to balance the interests. On one hand I had to advocate the interests

⁹ I didn't like this language and still detest it. It is contradictory in itself and this reflects even its definition. It's a language by hackers for hackers.

of the Vienna lab, on the other hand I saw the difficulties of mbp who were given extremely tight deadlines. Therefore I tried to moderate Vienna's demands. I found myself between two millstones and sometimes this was rather grueling. I could not be too hard on mbp since I had to keep up their spirit and dedication. I believe I succeeded with this balancing act since both the Vienna lab and mbp were satisfied with my efforts.

A few words about my colleagues from mbp who were all very clever and almost all of them mathematicians. N. CHRISTENSEN started as boss of the mbp-REXX compiler team. He was a cunning guy who liked managing more than technical work. He stayed in the team only for a short while. Oana KIPPS was a pretty, slim Romanian showing her womanhood and smoking like a chimney, sometimes stubborn but always open to good arguments. Working with her was a pleasure. Lothar SIEBERT was quiet and reliable. He had good ideas and had always a good overview of mbp's work progress. He became, very justly, CHRISTENSEN's successor. He was a man seeking balance, always hearing both sides and succeeded in doing so. J. ROBERTS, an Englishman always walking on sandals and all but fashion-conscious was an excellent computer expert. He had marvelous ideas, was rigorous in his work and nearly everything he produced was error free. If he ever made an error he corrected it speedily. After he had overcome our conflict he was an outstanding coworker with whom I got along very well. Over time he became the one who pursued successfully my ideas with mbp. Roger SPENCER, also an Englishman, was quite different. He was agreeable, dressed with sloppy chic, sometimes sloppy in his work but always ready to correct his errors straight away. His German was accent free and he was a brilliant "presenter" whose presentations were always interesting and never "slumberous". P. KÖRBER was – similar to ROBERTS – an accurate person whose products were nearly error free. He accepted without grumbling better solutions even when they weren't his own. Cooperation with him was a pleasure.

It should be mentioned that over all the years that we worked together, data such as design documents, programs, and test cases were transmitted over telephone lines from Dortmund to Vienna and conversely. This saved a lot of time.

The more we advanced in our development the more the compiler-specific difficulties became transparent. One of the problems, maybe the main problem was that a program generated by the compiler had to work exactly the same as when interpreted by the SPI. After all we had promised in our specifications that compiled and interpreted programs would be interchangeable. Because of that we had sometimes to include unnatural constructs. Therefore changes were necessary up to the end of the LLD¹⁰. As anticipated I was frequently at mbp and participated in

¹⁰ LLD: Low Level-Design

nearly all inspections. After initial skepticism mbp realized the value of this methodology and became enthusiastic proponents thereof. In order to participate in an inspection one has, of course, to know the material very well and I had to study incredibly much. On the train from Vienna to Dortmund I studied the mbp designs which were sometimes hard to grasp (in particular the cunning optimization phases). Sometimes an inspection's subject was rather boring (but important nevertheless). Then I sat with the mbp people in a conference room and had to fight an upcoming microsleep. But I succeeded and gained not only professional respect but also affection.

Engelbert, Klaus, and myself determined to produce commendable user manuals that could be easily understood. We cooperated closely with our "publications" people and were lucky that they usually followed our suggestions and contributed actively. The result was indeed exemplary books.

I forgot to mention our excellent test team. Walter PACHL, renown for his intellectual sharpness and his infallible feeling in "getting scent of errors". Heinz CHLADEK who helped us so much with the VM environment. Thomas KELLERMANN, a German assignee, whose positive criticism contributed a lot to our success. Andreas BRÜNDL, Eduard SCHEIBLER, and Wolfgang SCHRÖCKENFUCHS: they all tested intensely and perseveringly. They criticized but didn't work *against* us (what sometimes happens with test teams) but *with* us. If I remember correctly, they found close to 1000 bugs which we corrected before the compiler was finally shipped¹¹.

We worked like "Stachanowists"¹² because we wanted to deliver a quality product by the promised deadline. And we made it: In March 1989 we finished the first release of the REXX/VM compiler.

Delivery, however, was hanging by a thread. On the day before shipment we discovered an error in mbp's part of the product. We tried to fix it but couldn't. We called mbp and they provided some advice which, unfortunately, didn't help. Finally I had to call MOSER at 2 am (his wife was angry because the phone woke their daughter Barbara who had to give a concert the next day¹³) and asked him to come into the lab to discuss this critical situation (all professionals of the Vienna REXX team were still in the office). Critical because the mbp people refused to come immediately to Vienna in order to fix the problem. This I told Engelbert and he asked me to call SIEBERT in Dortmund (at 3 am) and to ask him to take the next flight to

¹¹ In contrast to current PC software where users are seen as involuntary testers, our REXX compilers were delivered without known bugs. Of course, this doesn't mean there are no bugs - no large and complicated program is absolutely error free.

¹² Term for Sowjet workers when the USSR was a communist country.

¹³ Barbara MOSER, Engelbert and Lisl's daughter was then an aspiring and is today an internationally renown concert pianist.

Vienna. When he refused, Engelbert took the phone and exerted enough pressure to get him to Vienna on the next morning. He found and removed the bug and the compiler could be declared finished. Our efforts and success were not left unrewarded. Every member of the REXX team got an "award" of a substantial monetary amount (I don't remember how much but it was certainly more than 10.000 Austrian Schilling, – about thousand US dollars). The awards were forwarded on the occasion of the yearly Kick-off meeting in the Viennese Marriott hotel. The Lab management presented us as an exemplary team and our good reputation spread over all other IBM software labs. Our compiler was an immediate success, also commercially. It brought in money since it sold very well. Our customers were enthusiastic and we received lots of praise and stimulation.

During the entire course of development we were in touch with REXX users within IBM who provided valuable comments and criticism. There were also heated discussions about features of the REXX language. The language wasn't formally defined but only by its implementation, the SPI ¹⁴. When there were unexpected results one did not know a priori if that was an implementation error or a peculiarity of the language. In such cases we posted the problem on a REXX forum in VNET and usually MFC, the language's creator, decided what the answer to the question was to be. When we, the Vienna lab, started to deal with REXX we participated actively in these discussions since we, as implementers, were interested in a stable definition of the language¹⁵. We Viennese people became soon known as REXX gurus since we treated problems thoroughly and more abstract than most other discussion participants. Even the godlike MFC who was a brilliant, conciliatory, sociable person, but stubborn, dogmatic¹⁶, and vain as far as REXX was concerned, had sometimes to accept our definitions. I believe that his sensational and well deserved success (which single person had ever invented a language that gained worldwide acceptance and moreover implemented it?) made him think to be infallible as far as REXX was concerned. Apart from these discussions many REXX enthusiasts were prepared to test our compiler before shipment (the so called beta version). This was valuable insofar as it complemented our own test activities. Again and again we found problems that would have been missed by our test cases. These contributions, our handling of problems and the discussions on VNET made our team popular, first within IBM and later with our customers. When a user experiences that his complaints are promptly and successfully handled he or she is happy to have helped

¹⁴ System Program Interpreter

¹⁵ In the late 1980s the American National Standards Institute established a committee in order to create a formal definition of the REXX language. Klaus HANSJAKOB was a member of that committee.

¹⁶ I remember two examples where he behaved vain and self-opinionated like a pope: The definition of the FORMAT built-in function which he – in my opinion – screwed up and refused to take anything back. The other was the definition of the precision in arithmetic that was brilliantly and correctly elaborated by Alfred GSCHWEND. Since this was contrary to MFC's opinion they quarreled for days in Vienna until Alfred refused to talk to Mike so that we had to use messengers between the two "enemies". It was just like international politics. I myself got along very well with MFC and we had many a drink together.

in improving the product and feels a solidarity with the team who made it. Our way to treat problems was seen exemplary. We never said "That's impossible" or "You must have made an error!" but we tried, usually successfully to keep cool, identify the problem, and finally fix the problem. If that wasn't promptly possible we told the customer how he could change the program to circumvent the problem. Our team never tended to say "This error can't be in my code, it must be yours!" because every one of us knew that he wasn't infallible and one's code was not necessarily error free. No-one was malicious or reproached the other about an error he had made. Again and again compiler users told us delightedly how much better the compiled programs performed when compared to their interpreted counterparts. Such praise flattered and tempted us to improve the next release¹⁷. We had collected pertinent ideas even before shipping the first release and it was now time to implement them. Immediately after shipment we organized a one week "Teach the Teachers" course for those system engineers who were to become REXX compiler specialists in miscellaneous countries. We explained thoroughly the compiler's structure and how to make the best use of it. Our "pupils", mostly experienced people were excited. Never ever have they had such a TTT course and our reputation was improved once more.

Our team was now the "showcase team" of the Vienna lab, role model for the others, known for excellent work and adherence to schedules. At the start of the project I had told Klaus HANSJAKOB that punctuality and reliability were of utmost importance in our team. Klaus agreed – to my relief (after all Austrians don't have the best reputation in that respect) – avidly because these were also his objectives. Soon there was no meeting where not all participants arrived on time. Engelbert accepted this our decision fully (after we once started a meeting without him because he wasn't punctual) and supported this rule.

Our punctuality became exemplary at a meeting which was to be attended by WALK, then the lab manager. When he wasn't there – as usual in sloppy East Austria – five minutes after the agreed upon time, we happily started the meeting without him. WALK entered later, looked puzzled, and Engelbert explained friendly but cool that we had started in order to save time. He would tell him later what he had missed. WALK was first baffled but then put a good face on the matter and praised us later for our very good working morale. Never again he was late for a REXX meeting¹⁸.

Now let me tell about the REXX team. I mentioned already that they were

¹⁷ From time to time (a quarter or half a year) a new release was shipped that had corrected the errors that were meanwhile found and had usually a better performance than the previous version.

¹⁸ Maybe I have a punctuality complex but I think that time theft – unpunctual people steal time – is a severe offence. Time can't be compensated and that not only in one's occupational life.

professionally excellent, willing to and capable of work and very cooperative at the same time. One helped the other when necessary and no-one was ashamed to ask for help. We had the ambition to deliver a product that was perfect in every respect and we identified us fully with the given task. Of course we were lucky to mix well. The atmosphere within the team was unbelievably good – in spite of all strain stemming from tight schedules. Among other things I made it a habit to enjoy the morning coffee together with all team members in my office. This became, over time, a ritual. "Duschi", a Yugoslavian¹⁹ girl, came with her coffee cart, each of us bought his snack and joined a conversation of about half an hour dealing with technical problems or philosophical, physical, or mathematical themes. Sometimes we just tattled. These recesses from work facilitated our togetherness and quite a few lab colleagues envied us about this arrangement. Some considered it an honor to be invited to this breakfast circle. I must mention that my office had a leather sofa²⁰, something quite unique at IBM Austria, that was ideal for such gatherings.

Another habit I pursued till my retirement was the following: When I couldn't find the solution to a problem even after extensive contemplation I asked a colleague (more often than not Klaus HANSJAKOB) to help me. I presented the problem as comprehensibly and precisely as possible so that my partner *had* to understand it. And very often (actually in most cases) I thanked for lending me an ear. The solution occurred to me while presenting²¹.

We were now a well-practiced team, the Viennese and the mbp people, and we were ready to take the next step after the second release of the compiler. This was to adapt the compiler for the other operating systems, MVS and VSE²². To that end we "imported" B.J.VANDEWATER (known as Bee Jay), an assignee from the US who knew MVS very well. He enjoyed his time with us so much that he later left IBM US and transferred to IBM Austria. Our team was further enlarged with Gerhard KAHLERT, a clever, imaginative, and modest guy who not only had built his own observatory but also played Jazz on an electric organ.

Another accrual was Sinan AKMAN, a young Turk, fresh from university who was to help me because my work load became too much: coordination of improvements and additions and the related inspections as well as problems with interfaces to MVS and VSE. All that was too much for one person. Sinan was intelligent, mastered German very well but showed from the outset little self-

¹⁹ The real, large Yugoslavia existed then.

²⁰ How did I, not being a manager, get to this status symbol? The former finance manager, Georges LESER, had purchased that sofa probably for Heinz ZEMANEK's office. When we moved to the IBM Building the rooms were smaller and too small for that sofa. I abstained from a second cupboard in my office and had thus room for the sofa which was well known in the lab and I was envied for having it. When I was very tired I simply locked my office's door and took a short relaxation nap. I recommend this to anyone.

²¹ An old experience: You learn most when teaching.

²² Successor of DOS/VS for "/370 Extended Architecture".

initiative. I attributed this to his youth and encouraged him to become active. I liked him and considered him my potential successor (as contact person to mbp). He prepared conscientiously for inspections and I let him lead them as moderator. Slowly he came in gear. When all inspections were completed and our workload decreased I decided to redesign the "options processing phase" since mbp had enough other work. This phase was actually rather simple: Testing the "option settings" for correctness and setting the corresponding switches, counters, and indicators. The old phase was already now in bad shape, hard to understand and confusingly written. Apparently it was written by SPENCER in passing, lacking a clear logical design. It consisted of many disconnected single routines. How could that happen in spite of our inspections? Over time there were always new options added the processing of which was just attached to the existing code. This conglomerate became difficult to maintain and I considered it necessary to replace it completely.

It was too late to rewrite this compiler phase and SPENCER's old one was kept (and is still in use, I believe). In spite of these difficulties the REXX/370 compiler was finished on 12 July 1991. This product turned also out to be an excellent one and was a big success with our customers. Meanwhile I got the honorable task to present the REXX compiler in the European IBM education center in La Hulpe, Belgium. I had to prepare a presentation of several hours, in English of course, to be given to a rather critical audience. Preparing that talk was no problem because I mastered the subject but I got somewhat nervous when I thought of my appearance. But when I took the stage my anxiety vanished and I performed my "show". The applause at the end told me that I was rather successful. One reason for that success was certainly that our product had internationally already a very good reputation.

The REXX project became second rank as money maker for the Vienna lab (the first one being the dignified and immortal DITTO) and the REXX team became a well-known and esteemed quality model. We were "shooting stars". Meanwhile IBM developed the so called SAA (Systems Application Architecture) and we had ambitious goals. We planned to implement the entire REXX language with the next release. So far we had left out some features because of complexity or time constraints. The most difficult – and most important one – was the INTERPRET instruction. And exactly this attracted me. I was already tired of my role being the contact man to mbp and implementing INTERPRET promised to be a task totally isolated from the operating system. Over time I had developed a disliking of the complicated interfaces to operating systems. From the outset of the REXX project I had not developed and implemented something real. I craved for tangible work.

When I hesitantly proclaimed my intentions, neither my manager nor my colleagues were enthusiastic, let alone the mbp people who treasured me. But I have had enough, I wanted to work on something real, I *wanted* to implement the

INTERPRET. My friends understood that I was qualified for that (at least they said so) and Engelbert released me. From 1 January 1992 I worked in Klaus HANSJAKOB's team and he became my "technical leader". Konrad KOCH became my successor and I taught him well and fast so that I could concentrate as soon as possible on my new task. This was to build an interpreter for the entire REXX language although the string given in the INTERPRET instruction was not allowed to contain all features of REXX. Klaus persuaded me to do so by arguing, rightly so, that the additional effort was justified in that all compiler test cases could then be used to test the INTERPRET instruction. The "only" remaining question was how to design the interpreter. First I considered compiling the string and then executing the generated code. This would have been elegant in theory but rather complicated in practice. I decided – approved by my colleagues – to build an “abstract tree” (=AST) from the given string and to interpret this by a tree walker. This tree walker starts at the root and walks the branches performing an appropriate action at each node. The tree walker was to be built by Regina PUCHER. The design of the AST I borrowed from that used in the compiler with the necessary deletions and additions. I worked closely with Regina in order to build AST and tree walker so that performance and storage requirements could be optimized. This design took a lot of time although I knew the principles from the outset: a “finite state machine” for the tokenizer and "linguistic functions" (as used in the SLANG compiler) for syntax checking and AST construction. My colleagues and my manager Engelbert were afraid that I wouldn't be able to complete this task in the given time frame. Indeed it was very much work but I enjoyed it just as much. I went to work happily as rarely (or never) before. When I had – to the surprise of the skeptics – finished the HLD²³ and got it inspected, I asked for support for LLD and coding. Gerhard KAHLERT joined me and was to assume the tokenizer and expression evaluation. Gerhard had never worked on a compiler and he was skeptical whether he could master this task. I reassured him and helped him in understanding the HLD and the underlying principles and let him loose. It turned out that I had found an ideal associate. He was clever, inventive, and rigorous. We cooperated splendidly.

The LLD of a complete Interpreter for a high level language like REXX is very elaborate and it is no wonder that Klaus and Engelbert were quite nervous when there wasn't any running code by mid 1992. But I knew that an impeccable and error free design is much more important than early coding. Early June 1992 we had the LLD finished and inspected²⁴ and coding could commence in /370 assembler language. Klaus and Gerhard wrote nice macros that saved lots of work. Since Gerhard and myself were old assembler shifters coding went on rather easily, and before my vacation in August we had already an interpreter that could process

²³ HLD: High Level-Design

²⁴ When I look today (1999) at the HLD- and LLD-documents of the parser I think they could easily serve as a base for a doctoral thesis.

simple statements. Engelbert and Klaus were reluctant in letting me take my vacation because they still doubted that we could finish the interpreter in time. I comforted them. Engelbert offered me a bonus, a "dinner for two" (at company expense) if I could get the SELECT statement running till 5th November 1992 (my 60th birthday). I spent a trouble free vacation without even thinking about work and came back loaded with fresh energy. A few days before my birthday SELECT was perfectly processed. From now on Engelbert had full confidence that we would complete the INTERPRET in time. Gerhard and myself succeeded easily. Of course there were some drawbacks but because of our clear design we could easily correct all errors we made. We sent (as usual) pre-versions of our new compiler including the INTERPRET support to the REXX community within IBM and offered prizes. Everyone who detected an error would get a Sacher-Torte. The one with most findings would get a trip for two to Vienna. Our hobby testers worked like crazy in order to win the first prize but there were fewer errors found than expected. Gerhard and myself were especially proud that no error was found in our interpreter. We got mails like "I tried hard to double-cross the Interpreter but couldn't! Congratulations!" We had finished our interpreter much sooner than necessary²⁵ and had now time to further improve the documentation and the program. I think we succeeded superbly. It should be possible to present our work piece to informatics students as example for pretty and good programming ²⁶.

Finally we compared our Interpreter to the SPI, the "official" IBM REXX interpreter. The result: except for extremely short trivial programs our interpreter was significantly faster, sometimes by an order of magnitude. We were enthusiastic and thought that based on these comparisons IBM management would discard the SPI and replace it with our interpreter. This would have the advantage of needing only one maintenance team for compiler and interpreter. This suggestion was rejected without telling us the reasons. This was our first disappointment. The second came when Gerhard had ported our interpreted to the system 6000, the fastest RISC computer²⁷. Very soon we had a working interpreter but our offering was again rejected. We started to doubt the technical competence of the higher management. Nevertheless we delivered "IBM Compiler and Library for SAA REXX/370" (the official name) punctually on 26 April 1993.

In spite of my excitement with work and colleagues I felt that I became somewhat tired. I wasn't inclined to adapt to the new computer world that could be seen on the horizon. A world which no longer valued quality programs but required fast delivery of badly designed and hardly tested products (have a look at current PC

²⁵ My longstanding experience told me that a clean thought-out design in the most important step in programming. Coding comes second.

²⁶ All that sounds like exaggerated self-praise but I have the documents that prove these claims.

²⁷ Reduced Instruction Set Computer

software). A world where one couldn't be proud of one's product because one knows that it is full of errors. A world in which my rigorous and scientific work style became an anachronism. A world where experienced, old professionals are seen as a burden. A world that appreciates only the "stockholder value". I accepted a RIO²⁸ offer.

With shipping the last compiler release my productive work for IBM came to an end. From now on I idled along which was tolerated by my manager and my colleagues. In December I had finished a report "Another Way of Organizing and Addressing Storage" that describes a storage architecture solving fragmentation and protection problems relieving the programmer from chin-ups and contortions. When I found out that LISP (a programming language of the 1960s "list processing") has used similar concepts I refrained from publishing my report. Nevertheless I believe that the concept I suggested is efficient and useful.

One thing I nearly forgot: Sometimes in 1991 or 1992 we were invited to supply a contribution for a book on REXX describing the REXX compiler. Although we realized that such a chapter could improve the popularity of our product no-one volunteered. Finally I did. Ten pages of the book "The REXX Handbook" (ISBN 0-07-023682-8, Mc Graw Hill) describe our compiler. The remuneration was remarkable: I got two free copies of the book. But I saw something I wrote for the first time in a published book. The cover of it stated that "world class REXX implementors" had written this book and I was among them. I don't know if the "REXX Handbook" became a financial success but Anneliese ISMAIL²⁹ whom I gave one of the copies was very taken with it and praised it. She says that she's still using it (1999).

Back to my last months with IBM. I prepared my leaving, distributed my treasures (technical books and stuff), helped my colleagues on occasion, was always prepared to advise, and tried not to be in the way of anybody. Engelbert was meanwhile promoted to second level manager responsible for REXX and DITTO. None of our group wanted to become manager (I had discouraged Klaus from it), so Erich MIKULA took over. I cannot judge his qualities since I was already in pre-retirement.

Looking back I can say that the REXX years were the most beautiful of my professional life. I had a most interesting project, was free and unbound and worked with a wonderful team of clever people. They delivered and accepted criticism, were cooperative, and became real friends. I couldn't have wanted better colleagues and

²⁸ Retirement Incentive Offer. With this offer IBM tried (successfully) to get rid of older, experienced, and highly paid employees. The fact that the company lost that way an incredible amount of technical know-how wasn't relevant to the company management. Ironically enough many of these retirees were later hired again as expensive consultants.

²⁹ The wife of my "twin" Moniem ISMAIL.

want to enumerate the members of the core team:

Engelbert MOSER (Manager)
 Klaus HANSJAKOB ("Technical Leader" of the "Run-Time-Systems")
 Sinan AKMAN
 Gerhard KAHLERT
 Willi PLÖCHL („Technical Leader“ of the „Compiler Proper“)
 Regina PUCHER
 B. J. VANDEWATER
 Mark TURNER
 Alfred GSCHWEND
 Klaus ZIMMERMANN
 Walter PACHL *)
 Heinz CHLADEK *)
 Günther SPANNRING *)
 Thomas KELLERMANN **)
 Lothar ZIRWES **)

*) Tester

**) Product-Assurance-people from Germany

13 August 1993 was my last day in office, then I went on vacation till 30 September 1993. From 1 October I was still an IBM employee but no longer active (I got 60 percent of my last salary). I had to be prepared to work again for IBM on request³⁰. On 30 November 1995 I retired officially (I resigned from the company, and now I get social security pension.)

³⁰ I was asked only once: I should have contributed something for MVS-VSAM. Convincingly I declared that I wasn't qualified for that and after that I was left in peace.