

Use of APL in Japan

Kyosuke Saigusa
APL Consultants of Japan Ltd.
1507 Shimomizo, Sagamihara
Kanagawa, Japan 229
Phone: 0427-78-2127
Fax : 0427-78-2205

Abstract

This is a historical account of the use of APL in Japan from the personal observation of the author. This is not intended to be scientific nor objective or based on studies or surveys of any sort. The description is mostly about IBM APL, because the author was employed by IBM for a significant part of his active years with APL. But the author also believes himself to be amongst the most informed persons in the area of the use of APL in Japan regardless of interpreter's manufacturers by being an open and independent APL consultant for 10 years now with wide contacts with various individuals in the world of APL.

How APL started in Japan

It was in 1971, when I was a staff-manager in charge of market forecasting at IBM Asia-Pacific Headquarters (APHQ) in Tokyo, my superiors, who happened to be a German and a Frenchman, told me to try to use APL in my data analysis work, because APL should replace desk-top calculators in IBM offices.

Soon, I was using APL\360 on the IBM 2741 typewriter terminal connected to the IBM System Center. It was probably the earliest use of APL in actual business applications in Japan. Earlier, in 1965, I remember hearing of a class on an Iverson Language held at the IBM-J System Center.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

APL 94 - 9/94 Antwerp Belgium
© 1994 ACM 0-89791-675-1/94/0009..\$3.50

Spread of APL within IBM Japan

In 1972, someone told me that an advanced version of APL, which featured file I/O, was being developed at the IBM Philadelphia Scientific Center. By that time APL had proven to be a very useful tool in our office.

Immediately it was arranged with IBM Japan to look into the possibility of bringing an early version of this advanced APL to make it the basis for the acceleration of the mechanization of IBM Japan's planning process.

This IBM internal version of APL, which was called APL-SV, really opened up a new horizon for us. At this stage we received very competent technical assistance from people like Joy Tuttle and Alex Morrow in Adin Falkoff's group at Philadelphia.

Prior to joining the IBM APHQ staff, I was a planner of management information systems for the Data Processing Department of IBM Japan. There, I studied in depth the internal logic of the MIS/360 and its boolean indexing mechanism in search of a good information systems.

In my new office, where I had strong management support, I decided to apply this experience to the new APL environment and eventually came up with a compact collection of functions, which made it possible for large corporate data files, such as Order-Entry and Inventory, to be accessed interactively by multiple users at the same time and data retrieved selectively, sorted or merged between files without physically moving records in the files, using the boolean index data prepared monthly when these files are downloaded to designated random access storage.

The concept was somewhat similar to the inverted file mechanism employed by "APLDI" released from IBM later. In comparison to "APLDI", this package could handle extremely large files in their original forms with the help of boolean index files created a relatively short

period of time on selected key fields depending on the requirements of different user groups.

As this 400 line compact utility package gained unexpected popularity in IBM Japan over GIS, MIS/360 etc., it was officially adopted as a standard end-user data retrieval tool by IBM Japan Information Systems. A group of users formulated automatically to assist each other in their problem solving. The package enjoyed the status of one of the indispensable tools to be used in the planning process at IBM Japan even long after I left IBM in 1983.

This tool was intended to encourage the use of APL primitives, rather than to cover them. I also made it educational for those who wished to write useful tools, or applications, not only in the economy of coding but in workspace design with the use of function libraries, naming conventions, portability, extendibility etc.

Beginning of APL marketing

In 1975 I was back at Systems marketing, IBM Japan, where my new mission was to market APL in Japan, based on our internal experience. The above-mentioned data retrieval system was made into a commercial package as CIP by IBM Japan and was named "APL Data Retrieval And Report Generation System" in February, 1976.

The first 17 APL customer sites were thus established within a relatively short period of time. Amongst them were Okamura Mfg.(Office furniture manufacturer), Mitsui Ship Building, NKK, Sumitomo Heavy Industries, Kubota Iron Works, etc.

We often collided with or resisted against the IBM marketing directions, such as VSPC as the only IBM program product offering the APL environment but announced initially without TSIO, which could have been the only external file access interface from APL.

At that time what I thought was the ideal APL environment was APL-SV. But APL-SV was not made into an IBM program product. I had to choose TSO as an alternative APL environment to promote APL because it was the only one that allowed an APLSV like operation under VSAPL. We relied very heavily on technical advice from Dick Dunbar and invited him to Japan twice to help in our marketing activities.

When the IBM 5110/5120 APL portable computer became available, our management did not approve us using these machines to demonstrate APL to DP customers, because they were not IBM DP division's products, and would steal the business opportunities for host

system VSAPL. Nevertheless, we continued to borrow and use these equipments to make effective APL demonstrations.

APL Symposia and early applications

In 1979, IBM Japan held its first APL symposium with Dr. K. E. Iverson as a guest speaker, thus we succeeded in gaining tremendous journalistic acceptance of this programming language. He gave various lectures in different parts of Japan for various groups of listeners. Before accepting our invitation, Dr.Iverson hesitated because, he said, he might say something contrary to what IBM would expect him to say to the customers. Anyway the invitation was finally accepted by the effort of my manager Toru Takeshita, who was one of the few authors of books on APL in Japanese.

Also in that year, in order to encourage more sophisticated usage of APL, System Center called upon all the internal skills to edit a consolidated guidance regarding APL installation, tools, methods and application development techniques etc., in Japanese. More and more Systems engineers showed interest in APL, which in turn quickly influenced IBM customers.

Significant applications were developed in various industry sectors, some of which were documented in papers presented at APL symposiums in Japanese in 1980, 1981, 1982 and onto 1986. "World Econometrics Modeling System" developed as a partnership program between Tsukuba University and IBM in 1981 was an example of APL applications development efforts in the early stages. This package is still maintained by my company for the Asia Economic Research Institute (a Japanese government organization), who participated in it's original development.

The Japan Automobile Insurance Rating Association switched from Fortran to APL entirely at that time and is still using APL. Japan Air Lines has been using APL in its Operation Engineering Dept. from this period. There are others such as Daiwa securities, Nippon Steel, Nippon Life, Yasuda Life, and Kyorin Pharmaceutical Co. in its Central Research Institute.

While the general mood was in favor of APL, the introduction of 3270GA in 1980 produced a strong effect of making people aware of new application areas of graphics programming under APL. Daiwa Security Economic Institute was one of such users of early APL/graphics applications based on 3270GA.

For myself, it was a great hint as to the possibility of incorporating Japanese language processing in APL applications, which had not been done anywhere else.

Soon afterwards IBM announced AP126 for the IBM 3279 color graphic terminal with a programmable symbol buffer. Based on this, I wrote a series of APL programming tools such as "Graphaid" (integrated cover functions for graphics/fullscreen/printer support provided under GDDM/PGF including Japanese character handling), "Textaid" (an experimental Japanese word processor with self learning dictionary) and "Diamond" (an experimental spreadsheet). These programs were distributed free, and later in 1983, after I left IBM, were compiled into a commercial package called "Application Development Tool" by IBM Japan.

APL Club and the growth of APL population

In the 1981 symposium, it was decided to organize a study group of APL users and to publish a regular periodical to exchange technical information. The periodical was named "APL Club" and in its 3rd issue a list of member names were printed in Japanese by APL, the first of its kind, and its APL program codes were also shown. This list, as of Dec, 1981, represented 113 installation sites (54 TSO, 33 CMS, 8 VSPC, 13 CICS, 5 unidentified) of 104 corporations (34 manufacturers, 12 banks, 11 Insurance, 1 security, 6 process, 8 distribution, 6 publishing/mass media, 7 government and private institutions, 4 utilities and transportation, 5 food and drug, 4 construction, 4 data servicers etc.)

The proceedings from the 1982 APL Symposium was printed in colors for the first time representing the trend for colors and graphics. NKK and Aishin-Warner presented manufacturing/design applications using APL graphics there. At the time when color was not quite a necessity, APL turned out to be a very good excuse for achieving the sales objectives of the 3279 terminals for the sales reps of IBM Japan. By the end of 1982, about 400 names of some 1000 large accounts for IBM Japan were listed as APL users.

Root of the mistakes in APL marketing

APL population growth proved IBM's marketing power rather than the real needs of the customers. I believe that IBM's motivation to emphasize APL then came mainly from APL being a big computing resource eater. Customer DP managers did not strongly object to this because it had the effect of enhancing their status in the company as they used more budget. IBM tried hard to squeeze APL into any DP installations by selling APL written packages, often, of inferior design, together with the concept of the "Information Center".

We all tend to ignore the very important characteristics of APL. It is a personal programming language es-

pecially for those who do not consider themselves programmers, but have a strong need to test their logic by computer. APL is not the language for commercial programmers who work by the hour or by the number of lines.

The only effective education on the proper use of the language is good live examples close to the users' problems, and the language specifications. The availability of experienced users to consult with is ideal. Skills should develop as required and in time, and not around the knowledge of the use of defined functions but by the appropriate selection and use of primitive functions at natural personal skills level and with the usage of peripheral functions via cover functions initially, but with increased understanding of the underlying logic at a later stage.

We must not forget also that real value of APL comes largely from what and how peripheral functions are used, and not so much from the language proper. People appreciate this language because it gives users the power to use not only the capability of processing complicated logic, but also the command of such peripheral functions as sophisticated data presentations, volume data handling, communications or even its operating system's environment etc., based on the knowledge of a single language.

In IBM Japan's marketing, very few academic and research institutions were approached. It was strongly biased toward business applications. Many large application development projects such as "EDSS" by IBM Japan eventually failed or seriously damaged the reputation of the language. I personally handled a case where IBM's "Application Prototyping Environment" almost ruined a large project development between IBM Japan and the former Mitsui Bank.

It mislead inexperienced APL programmers as to the design and the structure of an APL application. Since in a commercial application development environment, practically no developer pays attention to spending a large portion of development time to program tuning, redesigning after the functional requirements are fulfilled.

So many project failures are characterized by too many layers of function nesting, nested loops where matrix manipulation is a better choice, redundant codes coming from excessive modularization or formalization, immature use of general arrays where simple arrays can do better and the mismatch between the size of the application and the skills of the programmers.

As an example, an IBM program product written in

APL, which had some 150,000 lines of code, consisting of multiple workspaces, was eventually reduced to less than 5000 lines of code and a single workspace. I took the effort of doing this, because we had to include support for Kanji in the original package which was written for European languages. I recently studied the "PRINTWS" workspace distributed with the APL2/OS2, which uses "PMTOOLS". The whole design is far from the simplicity that we seek in APL.

Blooming of APL

After 1983, it looked somewhat promising for the future of APL. IBM Japan released Japanese APL developed jointly between the IBM Madrid and Tokyo Scientific Centers. We started using this from the second version which was released in 1985. We found this quite useful and developed many medium and large-size applications.

The Daiwa Securities' PMS/PC, which is available as a commercial package written in Japanese APL, is a very handy and flexible graphics presentation system of time series data analysis for Japanese stock buyers. This is a miniaturized version of Daiwa Security's "Portfolio Management System", which is also written in APL2 under TSO. The monthly data updating service is still done.

An integrated accounting analysis and report generation system for Mitsu Marine and Fire Co. is one of the main applications for the controller's department, and based on the analysis of original accounting data transmitted from the host system to a personal computer running under DOS, it produces most of major reports in Japanese such as the Profit and Loss statement, Balance sheet, reports to the Ministry of Finance and the Board of directors and stockholders etc.

APL there is used not only for computation, but also to manage limited computing resources for the system. This system was converted to AIX/APL last year because it finally exceeded the limitations of resource constraints due to the increase of data as well as frequent modifications and additions of features. This is also an example of a self maintaining system and it is designed to adjust its processing logic and report formats automatically where possible as new data items are added. Thus maintenance and running costs are minimal.

The next three examples are for Morinaga Dairy Products Co. The first of these is a management information system on a daily, weekly, monthly and annual basis and presents various management data in graphic and table formats on PC displays connected to the host data base

via ECF. Interactive and batch modes and On-line and Off-line modes are provided for the most efficient data presentations. It was originally converted from three independent information retrieval systems at management and operational levels, all written in VSAPL under TSO. This was done to reduce data processing cost, but originally no one believed that only a small space under DOS can accommodate APL, Graphics and communications on top of applications. For the top executives of the company to use, it was designed to operate on fast data scroll technology without the need to enter selection data from the keyboard. This is also running daily.

The second one is a comprehensive material test system. All the material to be used by some twenty production plants throughout the country are tested in the central laboratory before permission of use is issued. The entire process is managed by Japanese APL running under Japanese DOS on a nation-wide computer network. This is a vital system for the corporation.

The last one is an ice-cream formulation system. From the composition of the Ice-cream mixes and products to the issuance of production orders with legal labeling text printouts, the system covers the most important aspects prior to production by its own and subsidiary companies' ice-cream plants throughout the country.

There were many other systems, which were more or less experimental until actual requests should come in such as the health insurance accounting system for hospitals using scanners at doctors' desks and a Cadam-like system model which accesses DOS graphics driver directly from Japanese APL etc.

Decline of APL popularity

Outside this, between 1983 up until recently, it is the history of the decline of APL. So many software service providers selling APL skills appeared and disappeared during this period. So many APL users ceased to use APL. So many APL business attempts such as Ampere's laptop machine and Sord's APL personal computer, Hitachi's APL interpreter, Fujitsu/Toshiba's interpreter failed to realize their expectations. A lots of complaints were heard about APL locking up the entire system from DP people, who openly showed hatred toward APL.

IBM began to advise APL users to convert to non-APL packages such as AS. IBM no longer showed interest in keeping APL users current with the latest developments in APL interpreters. They stopped to issue APL club, and the APL study group of IBM users dissolved. APL symposiums weren't held any longer. IBM Japan did

not join Europe and America to announce APL2/PC, when it became available. There is no way of knowing why, but the result is the Japanese essential requirements for the support of DBCS (Double Byte Character Set) was missing from the APL2/PC.

DOS/V and APL2/PC with DBCS support

Until the end of 1990, when DOS/V was announced, Japan was isolated from the rest of the world in the area of personal computers. In order to support DBCS by hardware, even IBM PC in Japan was not compatible with IBM PC for the rest of the world. They had to use a special DOS which was different from standard IBM DOS. Japanese APL ran only under Japanese DOS on the Japanese PC. This were expensive too, with even a low range PC costing between 1 to 2 million yen from IBM. They were found only in the offices of big companies. The majority of the PC's were from NEC, which did not allow APL to run.

DOS/V implemented DBCS support by software based on the standard DOS used worldwide. By this IBM Japan had to maintain both J-DOS and DOS/V. IBM Japan started selling lower price PC's, which no longer supported old J-DOS, hence J-APL.

This was the beginning of a revolution for the personal computer world in Japan. The prices came down and power increased very rapidly. Compaq, Dell, and many Taiwanese IBM compatible machines rushed into the Japanese market. J-DOS packages were converted to DOS/V. When I was asked by IBM-J if it is necessary to convert J-APL too, I had to answer "no", and told them that it is not the time for APL-1 and that they should consider announcing APL2/PC with DBCS support. IBM Japan listened the first part and not the second part.

It was really a crisis for my business too, because not only many applications had already been written in J-APL, but also this language was used in various useful purposes including conversion from host VSAPL applications into APL2/370 etc. for companies still using APL. We had also possessed many utility functions including auxiliary processors, and text books. We had also invested heavily in Japanese PC from IBM Japan, which no longer ran with DOS/V. If we stuck with the old J-APL, we would have to close our business sooner or later.

So I purchased the first IBM compatible machine from my old friend from IBM, Charlie Wilks, when we had the APL conference at Stanford. We immediately started to investigate if it is possible to modify

APL2/PC to make it to run under DOS/V and to have the capability of handling Japanese characters, so that we can convert past applications written for J-APL.

In converting, a greater advantage was expected from various new features of the APL2/PC. We found it was not possible for us to make it completely compatible to J-APL in the area of DBCS handling without access to the interpreter's source code. Besides, it would have been beyond our means even if the entire source code became available to us.

Finally we decided to choose the secondary option, which allowed us to display and input Japanese characters on AP124 designed screens only. Thus application users couldn't see the difference at all. We had only to give up session managers and some programming facilities. Before this work was completed, the person in charge of this task had to leave us to inherit his family business. So I was left with this big task without any professional skills of programming in DOS assembler. A kind American friend offered me a rescue. So by the end of 1992, I was not only able to convert any of J-APL applications into APL2/PC, but had developed many useful utilities, such as an APL general-purpose editor using the new AP124.

This editor, which switches between Japanese and English display mode by DOS BIOS calls issued from APL, allowed entry and display of Japanese texts together with APL characters. It also provided a convenient option of entering APL characters both from the keyboard and the screen when keys are not marked with APL symbols. In addition, it provided quite easy cut and paste capabilities and many other features which I needed and were often not available in similar editors. Besides it was written in APL for any APLers to customize or extend. At the same time all the APL programming tools that I had ever written were reviewed in the new environments including HOST and AIX and were rewritten entirely at much higher interactivity and uniformity throughout those environments.

We already had experience in producing an English version of Daiwa PMS/PC using APL2/PC and presenting it at the 1989 APL conference in New York in the vendor's corner. But the first Japanese application, based on this APL2/PC-J, was for a dealer of Shimazu Manufacturing to use to produce sales proposals and specifications including the attachment of engineering drawings in A3 format. The application is also an interplay between the sub-notebook that the salesman carries and a desk-top PC at the office. The sub-notebook programming was done in Japanese APL DOS/V version. This last product was another work of mine.

Japanese APL- DOS/V version

Toward the end of 1992, IBM called me and asked if I could convert Japanese APL to DOS/V. And they told me that they succeeded in persuading Mr. Shiina, who was retiring as the president of IBM Japan to grant a small fund so that he put a period to the hustles and bustles of APL while he was in office. The intention of IBM was to try to revive the popularity of APL before the expected announcement of APL2/OS2. This I accepted, and spent 4 months January thru April of 1993 to complete the conversion with some extensions for a VGA graphics auxiliary processor.

This was the second time I programmed in DOS assembler. This was distributed in the nationally popular PC magazines as a freeware. At the same time, it was uploaded to the Nifty Serve freeware library, from where some 800 copies were taken in a very short period of time.

To test the quality and usability of this product, I converted one of the business consultation applications used by the salesmen of IBM dealers throughout the country from J-APL with very good results. Recently the Automobile Insurance Rating Association asked us to help to improve and add new features to an APL application written by themselves many years ago in Japanese APL. This was done using this freeware, because the internal code, the workspace as well as APL data files are 100

I am beginning to discover that numerous small applications exist and are being used daily in very unexpected places and by unexpected people. APL used this way seems to have spread from the individuals who once had a chance of using APL under the Host system environments.

Some years ago, when I was in Osaka at Nippon Life Insurance working on modification of their giant system of profit analysis written in APL2 under TSO with interfaces to COBOL subroutines and DB2 access, I had a call from a person from Nara, 30 kilo meters away from Osaka. He introduced himself saying that he worked for a small manufacturer in Nara, and he had learned to use APL when he was with IHI in Tokyo years before. I did not ask how he found me, but he was very satisfied, when his trivial questions regarding the use of APL were answered.

A professor of a university recently visited me from Takarazuka near Osaka to seek contact with me. He told me that he had been named as chairman of a committee to study the Japanese Labor problems under the auspices of the ministry of labor. He left with me a report from the committee with a lot of graphs and ta-

bles. He told me they had been produced by himself using Japanese APL.

APL/AIX

In 1991, an IBM sales representative for the Tokyo Power Electric Co. called me and asked me to take care of the expected conversion of planning applications written in VSAPL with GDDM graphics into APL/AIX when it was released from IBM. This was to be the first account in Japan to use this new interpreter. I thought the Unix machine was a little too expensive for us to buy, but they arranged so that we could purchase RISK/6000 model 220 at good discount price. We started using a pre-release version of the new APL interpreter. We had very little knowledge of the AIX environment as well as the use of its commands. IBM had assigned a Tokyo Scientific Center staff member to the task of clearing all the technical difficulties of introducing this new interpreter to the Japanese market environment.

Initially we were misguided by the IBM documents which clearly stated that there was no support for the Japanese. IBM-J had to develop an interface program to input and display Japanese texts in C. This approach looked very unnatural and impractical for the Japanese users. One day when we brought in an application from APL2/PC and used it in the Japanese window it worked without any problem in displaying Japanese texts. It did not present any serious problems in accepting Japanese text entry.

Soon we found that this is a very good environment for APL programming/testing and application operations. At the same time we found APL2 can be used on any platforms and the product can be transported across the platforms including Japanese with the understanding of some negligible limitations.

Some work had to be done, however, on AIX. That was when we had to develop utilities to support various types of Japanese printers, especially for Kanji and graphics, or bit map manipulation, where AIX often failed to behave as we expected. With this we were able to develop most of the conversion tools including the correction of the migrate workspace provided with the system, because this workspace only partially considered the DBCS environment, thus it is entirely useless for the majority of Japanese users migrating from the APL2/370 or VSAPL to APL2/AIX.

In this way we ended up gaining a position close to monopoly in the area of workspace conversion into the AIX environment. A large part of the conversion work

consists of the analysis of the existing workspaces, and every aspect of this is carried out by writing instantaneous APL programs. We cannot perform this work without the use of APL2/PC now.

Last year Japan Air Lines Operations Department decided to stop using the IBM system which was dedicated to APL and Fortran. The entire system, some 40 frequently used APL workspaces, 80 Fortran source programs and numerous related data files amounted to about 150 of 2HD floppies. These were all converted into the AIX environment within about 40 days.

APL conversion took only one person and a few days because they didn't use Japanese, but Fortran conversion took several people and nearly two months, including rewriting of EXECs into shell scripts. We are now talking about running fortran under APL for the ease of system maintenance. These fortran programs are supplied with the aircraft when they are purchased from the aircraft companies and must be used as they are given.

Another case of conversion was for a company called "Japan Port and Harbor Consultants". They were using APL in various forms throughout the company. The main applications were on the IBM 9370 and APL2 under VM/CMS. Japanese APL was also used very extensively. Main accounting applications were processed on the Ampare WS1 using APL/68000, using 3 APL workspaces written personally by the former president of the company. I was not aware of the existence of such an APL user group until I was asked to consult on this conversion project.

The next case of conversion, is a central lab of the Kyorin Pharmaceutical company. They have been using the IBM 4390 dedicated to APL and Fortran, which they are replacing by a dozen of IBM RISK/6000. They use Fortran link, DB2 link and Japanese texts. The details of applications are confidential, like many similar APL users, but they say the laboratory operation depends very heavily on the use of APL.

While I was writing this, I had a call from Yasuda life insurance for consultation on the conservation of CPU resources on the Host systems when APL runs. I would expect to discover many more silent users seeking support which they cannot obtain from IBM any longer, when down sizing progresses in these years of economic recession. I also think that these survivors of the hard age, could become the basis for the revival in popularity of APL in a much more sensible way than before.

Estimation of APL penetration today and in the future

My wild guess as to the size of the usage of APL in Japan is about 400 IBM mainframes, 50 Hitachi/Fujitsu mainframes, at least 2000 individuals on Japanese APL, at least 6 APL2/PC and many more illegal users, and an about 20 companies but growing figure on workstations. IBM Japan finally announced APL2/OS2 with DBCS support for the Japanese market. There is some doubt whether it will succeed, because more companies have plans to go to Windows rather than to OS/2. Many windows applications do not run under WinOS2-J. Many adapters such as fax and graphics are not supported by OS/2 either.

On the other hand, however, there is a sign of OS/2 gaining increased popularity in Japan as the prices of high-end PC's are coming down rapidly.

IBM Japan should realize the importance of APL2/PC with DBCS support which will continue to provide excellent application environments in terms of economy, performance, extendibility, flexibility and all. They have already deprived Japanese business specialists, analysts, engineers, programmers, researchers and students of the opportunity of applying the power of this language in the PC environment that they already possess.