THE STANDARDIZATION OF PROGRAMMING LANGUAGES

Franz L. Alt
National Bureau of Standards
Washington, D. C.

The economics of standards

Standardization in the computer field has been under discussion for at least ten years; and for the last four or five years, concrete efforts in this direction have been under way, centered in the American Standards Association. Yet even today there is no clear-cut answer to the question whether or when or to what extent standardization in this field is desirable. Work is being done on standardizing of both hardware and software; and in the latter area, it ranges from the smallest units--characters--to the largest--programming languages. Why should there still be so much difference of opinion on the value of standards?

As a staff member of the National Bureau of Standards I know from daily experience that in most fields of technology standardization is an unmixed blessing. Things are not nearly so simple in the field of computer languages. Whatever the advantages of standardizing one or a few computer languages, we pay a heavy price for it. But we also pay a price for failing to standardize, or standardizing too late. And the heaviest penalty of all is for standardizing prematurely or on the wrong language. Between these conflicting tendencies we have to try to reach an optimum compromise. But what is optimum depends on the viewpoint. There are a few persons who are flatly opposed to any standardization of programming languages at this time. Many more agree that some amount of standardization is now desirable, but in varying degrees and with varying safeguards. What seems best to the user does not seem best to the producer of computers, and what is best for one particular company is contrary to the interest of all other companies.

Objections to standardization

The languages which exist today are by no means the best possible. We do not yet have a good theory of computer languages, and we are nowhere near the limit of the concepts which can be expressed in such languages. Existing languages are constantly being improved upon. If everyone agreed on using exclusively some standard programming language or languages, the development of new or improved languages would be stifled. Persons with ideas for improving languages would be discouraged by the knowledge that the results of their work would not be used for a long time, and then only after the cumbersome process of revising an existing standard. The situation is quite unlike that in, say, screw threads, where revolutionary new inventions are hardly to be expected. The desire to encourage further development of languages is so strong that it has caused some people to object to any standardization at this time. While this viewpoint seems to me to be too extreme, it does seem necessary to strive to conduct standardization in such a way as to minimize its harmful effect on innovation.

Another point is the great and growing specialization of computer languages. It would be wholly unrealistic to imagine that all computer programmers could agree on using one and the same language. There is an increasing tendency to narrow the fields served by a language, to make the languages more and more specialized. What we need, therefore, is not one standard but a number of different standard languages for different purposes. There will unavoidably be overlaps in some places, and gaps in other places. It is easy to see what a difficult problem it will be to choose a complete, consistent and non-overlapping set of programming languages.

From what I have said so far, two limitations on our standardization effort emerge: There will not be one standard language but a collection of them, so as to accommodate all areas of widespread machine use. And it seems advisable that the use of the standardized languages be optional and not mandatory, so that nonstandard languages may be used or newly created whenever this appears to be economically justified.

The case for standardization - different viewpoints.

What makes standardization desirable at this time is the proliferation of languages which has begun to set in during the past few years. This threatens to limit the possibilities for interchange of programs between laboratories, and reduces the rate of return on investment in compilers. There is now a bewildering variety of different languages, often differing from each other only in minute detail. Sometimes this comes about merely because programmers insist on their own way of doing things, at other times because of their understandable desire to create a language best suited for their particular brand of problems. Sometimes it is to facilitate the writing of compilers for a specific machine, or to take better advantage of some special feature of a machine.

And here we can discern the germ of disagreement among interested parties: Different languages are best suited for different machines, and if one language is chosen as a standard, this gives an advantage to one kind of machine. True, it is a very temporary advantage, for computers become obsolete very quickly, and new models are introduced every few years. Once a standard language has been agreed upon, manufacturers will usually find it possible to adapt their next model to that language. In a few years everyone would be on an equal footing except for the fact that standardization itself is not standing still, and we may expect new languages to be added to the list of standards from time to time.

There is another way in which standardization affects the competitive position of companies. A computing laboratory which intends to switch to a new machine has a library of programs written in a language adapted to its old machine. Therefore, it has an incentive to acquire a machine of similar characteristics, presumably by the same manufacturer. Also, many laboratories are interested in exchanging programs with other installations; there is, therefore, a tendency to acquire a kind of which numerous others are in use, or may be expected to be in use in the future. Thus, once a manufacturer has acquired a large share of the market, there will be a tendency for his share to grow even more. The only way in which the others can compete at all is by providing compilers for the languages used by the dominant producer. This is made more difficult if there exists a variety of such languages, differing in minor points. Another road open to the small manufacturers is to provide compilers for a language which is common to all of them but different from that of the dominant producer. This may make their combined slice of the market large enough to be competitive. But it forces them to invest in producing still another compiler for each of their machines, and this investment will pay off only slowly.

In any event, those who buy or rent computing equipment have acquired the habit of expecting that the producer will furnish compilers for several popular languages. Since each customer has his own desires, and since even the major languages exist in several slightly different versions or dialects, the manufacturer is hard put to satisfy all his customers. A consequence of standardization will be that more customers will be inclined to expect compilers only for those languages which have been adopted as standards, thus reducing the pressure on the manufacturer. This is the reason manufacturers support standardization, and it is also the reason why they will in general favor a small number of standards. In this respect the situation in programming languages is similar to that in many other industries. The customers, in general, will argue for a somewhat greater variety of standard languages, to facilitate programming and the exchange of programs in a greater number of special fields; but even the customers have an interest in holding the number of standards down to some reasonable level, since otherwise opportunities for exchange of programs are too much reduced.

Dangers of premature standardization

Matters in this respect can perhaps best be illustrated by comparison with an entirely different field, that of units of measurement. The American economy today is significantly handicapped by its adherence to the English system of units, while most of the world is using the metric system. In the eighteenth century, England was the only nation in which industrialization had gained a foothold. This required relatively precise standards of measurement, and the English units naturally were chosen as standards. At the time of the French revolution the superior metric system, then newly devised, was introduced on the European continent without opposition; but it was rejected in England, where it would have conflicted with the vested interests of the older industry. The American colonies, after independence, switched to a decimal system of money but stayed with the mother country in the matter of the physical units, the foot and the pound. As recently as 1914 the lion's share of world trade fell to countries with English weights and measures, but since then the share of the "metric bloc" has risen rapidly, and now our adherence to English standards constitutes a severe impediment to our export trade. But the enormous investment in our industrial plant makes a changeover extremely expensive.

It is too early to be sure, but we may discern the danger of a similar situation arising in programming languages. In the early years, and even today, computer development in the United States has been far more rapid than in the rest of the world. We were thus the first to develop higher-level programming languages, and we are now saddled with a sizable investment in computer programs written in the early, and, therefore, less perfect, languages. Some day this may cause us to fall behind in comparison with the rest of the world.

Both producers and users are interested in standards which have international validity. For the producers, the reason is obvious; they wish to compete in the foreign market. For the users, the reason is less obvious, but actually even stronger. Although the volume of computer activity in Europe is small, the technical level of contributions to the theory and practice of programming is very high. The problems which programmers have to solve are, with few exceptions, the same in all countries. Interchange of computer programs across national boundaries, while still small in volume, is a practical thing today.

I have gone into the situation in so much detail in order to give an idea of what conflicting pressures we may expect, and particularly in what ways the field of computer languages differs from other industrial fields in which standardization is being attempted. I now propose to discuss first the form in which the American Standards Association has organized its effort for programming languages, then the present status of specific languages, the contributions of other organizations, and the outlook for future accomplishments.

The standarization effort of ASA.

Subcommittee X-3.4 of the American Standards Association, charged with the standardization of common programming languages, is less than four years old. Like many other ASA committees, it has set up a number of working groups which deal with detailed technical preparation of various items on the committee's agenda. The full committee meets at intervals of one or two months, considers and discusses the reports of the working groups as well as some items of business coming in from other sources, and recommends items for action to its parent committee X-3. The working group structure, in other words, the system of distributing tasks among the working groups, is not based on one logical principle, but is a hybrid of two principles. This situation has come about historically. It was thought originally that different working groups should attend to different aspects of the standardization process, each working group being assigned one aspect for all programming languages. Later on a need was felt for having one working group in charge of all the work on one particular programming language. As a result we now have some working groups concerned with individual languages, some with particular functions, and some with assignments of both kinds.

In addition to the working groups, who are at present eight in number, there is a Membership Committee, which screens applications from individuals who wish to become members, alternates or observers, and a Steering Committee, whose function has been compared to that of the Rules Committee in Congress. The Membership Committee also proposes the general rules which govern the admission of individuals, subject to approval of the main committee.

The voting members of X-3.4 serve as individuals, not as representatives of their companies or agencies. Furthermore, we have a rule that there can never be two or more voting members who are on the staff of the same company or agency. Additional persons from the same organization may, however, be admitted as alternates or observers, and as such take part in the debates, but without voting. Otherwise a person's technical qualifications and legitimate interest in problems of language standards are the only criteria for admission.

Working Group X-3.4.1 is concerned with the theory of languages and language standardization. It is thus a purely functional working group, not concerned with any particular language. It made most of its contributions in the early stages, by specifying the process of standardization, criteria to be applied at different stages, etc. It has produced or stimulated a large number of publications and sponsored or co-sponsored technical meetings. It has made studies of language structure and surveyed the existing organizations concerned with computer languages. At present it acts as a stand-by advisory group which we consult whenever a theoretical question comes up; this happens more often than one might think.

Working Group X-3.4.2 has several functions. First of all it is concerned with language specifications, i.e. with the way in which a language is to be described. This is a major problem, since the description of any one language is likely to fill a book. Next, X-3.4.2 is concerned with establishing criteria for the selection and evaluation of languages. It looks at a number of existing programming languages and decides which ones might make good candidates for standardization. If a language looks at all promising, a subgroup is set up inside X-3.4.2 which examines that language in more detail and arrives at a recommendation as to whether the language should indeed be considered in depth as a possibility for an American standard. If the recommendation is positive and if it is accepted by the parent committee, a new working group of X-3.4 is set up, outside of X-3.4.2, to deal with this language exclusively, and the responsibility of X-3.4.2 for this language ceases. The working group created for the language then concerns itself with all the detailed questions of which features are to be included in the standard and how they are to be described, and comes up with a document intended as a draft standard. This document is referred back to X-3.4.2 for comparison with its own criteria for language specifications. In this way we hope to achieve some uniformity both in the selection of languages for standardization and in their description.

Working Group X-3.4.3 is devoted entirely to one language, FORTRAN. Its accomplishments will be surveyed later in connection with discussion of individual programming languages.

Working Group X-3.4.4 has two assignments, one functional and one concerned with a specific language. This group is charged with the investigation of standards for processor specification, both hard and soft. In addition, as their initial effort, they are pursuing the standardization of COBOL. It is recognized that this in itself is a subset of their major responsibility and it is believed that the experience gained from the COBOL standardization effort will contribute substantially toward subsequent efforts in the general area of processor specification. In fact, this working group has spent most of its efforts on COBOL. Its progress will be discussed below.

Working Group X-3.4.5 is concerned with international matters. This is entirely a functional assignment. It includes (a) relations with the international standards organization (ISO); (b) backing up the American delegation to meetings of ISO Technical Committee 97, Subcommittee 5, which is concerned with programming languages; (c) in effect, Working Group X-3.4.5 has served as the Secretariat of ISO TC97 SC5.

Working Group X-3.4.6 is concerned with nomenclature; this is entirely a functional assignment. The group cooperates with Subcommittee X-3.5 (Glossary) by furnishing suggestions and criticisms on glossary items having to do with programming languages.

Working Groups X-3.4.7 and X-3.4.8 were organized just recently. They are concerned with APT and ALGOL, respectively. Previous to their establishment, a good deal of work on these languages was done by subgroups of X-3.4.2.

The present status of specific languages

In its almost four years of existence, X-3.4 has not yet produced a single concrete standard. This bald statement may cause some shock, and may further lead to the suspicion that the conflicting pressures to which I have alluded have tended to prevent the selection of any standards. To the best of my belief such a suspicion is entirely unfounded; the true cause for the long time lapse lies in the fact that a programming language is an incredibly complicated thing. It is probable that nothing even approaching it in complexity has ever been standardized. A language is far more than a list of macro-instructions together with a compiler which translates them into machine code. There must be, for each macro-instruction, a description which specifies what output the computer is to produce on receipt of this macro-instruction in all conceivable circumstances. The action which the computer is to take may depend on previous instructions, and there is an unlimited variety of ways in which instructions can be chained together. It is a difficult problem indeed to ascertain that all cases have been provided for and that no contradictions will occur.

In fact, it has been found repeatedly, when a language was being investigated as a candidate for standardization, that no satisfactory description of the language existed, or that the language itself or its implementation was deficient in serious ways. In such cases, development work is needed before standardization can take place. X-3.4 has taken the position that the work of developing computer languages, as distinct from standardizing them, is not its responsibility, nor a responsibility of ASA in general, but should be left primarily to the professional societies and other groups.

But Subcommittee X-3.4 and its working groups are actively cooperating with other groups in such development work, and are furthermore carrying on development work alone in cases where no other organization can be found for it.

To date X-3.4 has concerned itself with four specific languages: ALGOL, FORTRAN, COBOL and APT.

The standardization of FORTRAN is the object of a special working group, X-3.4.3. The group decided early that it would be necessary to standardize two different versions, similar to what is known as FORTRAN II and FORTRAN IV, so that the user will have a choice of two levels of language, one narrow, with few instructions, but easy to implement and compile on computers of moderate capacity, the other more comprehensive, allowing greater power to the programmer but more difficult to implement. The two standards are to be "upward compatible" so that a program written in the narrower language can be used with the compiler intended for the broad one, but not vice versa. Draft documents describing both languages have been prepared, copies of the tentative drafts have been distributed to interested parties, and their comments have been worked into the drafts. We are looking forward to early submission of these documents as proposed American draft standards. Furthermore, on the international level, the International Standards Organization is awaiting a U.S. FORTRAN proposal in the hope that an American standard may be agreed upon, which may be acceptable as an international standard without changes. It now seems likely that they will soon be presented with a recommended draft.

The effort on ALGOL was in the hands of a subgroup of X-3.4.2 until recently, when a separate working group X-3.4.8 was established for this language. There is strong interest in ALGOL in Europe, and so the International Standards Organization has taken the lead in working on an international standard for ALGOL, while the American preference has been to wait for this to happen and then perhaps to adopt the international standard for the U. S. without change.

At the time when these strategies crystallized, two or three years ago, the most comprehensive version of the ALGOL language then in existence, known as ALGOL-60, had at least two major faults which, in the opinion of many, made it unsuitable as a standard. First, there appeared to be some unresolved inconsistencies. Second, no input and output instructions were defined; this was left to be done by the compilers. The latter objection was removed when a subcommittee of ACM, with the cooperation of X-3.4.2, developed a set of IO specifications. As a result, it was decided in May of 1964 to draft standards for ALGOL on two levels, with upward compatibility: one being the full ALGOL-60, the other a subset sufficiently restricted to remove the inconsistencies of the former. The standards are also to incorporate two levels of IO specifications, based on the ACM report and on work done by IFIP on the problem. These decisions are so detailed that no more than an editing job is still required to create a document suitable to serve as international standard.

COBOL has been assigned to X-3.4.4 along with the general problem of processor specifications. The difficulty has been that there is as yet no document in existence describing the COBOL language in a manner sufficiently clear and unambiguous for standardization. To accomplish their work in COBOL standardization, X-3.4.4 is subdivided according to four functions: (1) A working group to conduct COBOL implementation surveys. Work on the first survey is nearing completion and reflects the details of all COBOL 1961 implementations in this country. The purpose of the survey was to suggest the largest intersection of language elements to be included in the first level standard. (2) A working group to specify a set of COBOL programs to be used in order to determine whether a given implementation complies with the standard. These programs are initially designed to test the availability of a given language feature and subsequently will determine the validity of the final result. In no way are the programs intended to measure processor efficiency. (3) A working group to publish the X-3.4 COBOL Information Bulletin. The purpose of the CIB is rapid dissemination of information on COBOL standardization activity. To date four bulletins have been published and a fifth is currently being completed. (4) A working group to prepare the proposed draft American COBOL standard and its approved subsets. This group is currently working with the CODASYL COBOL Committee in a cooperative effort to develop and standardize these subsets.

APT is the most recent addition to the list of languages selected for standardization, and the preparatory work is not yet completed. A subgroup of X-3.4.2 had been at work for some time before the recent formation of a separate working group, X-3.4.7, which will now concern itself with the creation of a draft document.

At the same time the committee continues to consider other languages which seem promising for standardization.

Other standards-processing organizations

While the American Standards Association has carried the largest share of the standardization effort, other technical organizations have contributed significantly. Although these organizations have widely differing structures and purposes, there has been a remarkable absence of jurisdictional disputes, overlap or other confusion which one might fear. On the contrary, all has been harmony and cooperation in regard to organization; what disagreements exist pertain to technical or economic matters, as they ought to. This happy state of affairs is due in great part to the fact that the same few individuals serve in all these organizations.

Association for Computing Machinery.

The Standards Committee of ACM cooperates in every way possible with X-3.4. Many of its members are also members of X-3.4. Its chairman is also editor of the Standards section of Communications of the ACM, which performs a most valuable service by keeping interested parties informed of every step in the standardization process. Various Subgroups of the ACM Computer Languages Committee are active in language development. This activity is of importance to standardization since in many instances the lack of adequate development is what prevents us from standardizing. In particular, the ALGOL subcommittee of this group has worked closely with the ALGOL subgroup of X-3.4.2 in developing input and output specifications for this language.

International Federation for Information Processing

This group (IFIP) and its U. S. adherent (AFIPS) take part in language development and thereby contribute greatly to standardization. AFIPS, or rather its predecessor, the Joint Computer Conference Committee, was probably the first organization in which standardization in the computer field was seriously discussed. IFIP has a number of technical committees, among which TC 2, concerned with programming languages, is of interest to us. In particular, IFIP-TC 2 has a working group for ALGOL, which has been actively engaged in the development of this language and deserves the principal credit for the formulation of the "full ALGOL" standard and one of its IO systems. Close cooperation between IFIP-TC 2 and X-3.4 is assured by overlapping memberships, a number of individuals holding positions in both organizations and contributing decisively to their work. Another technical committee, IFIP-TC 1, is concerned with the standardization of computer terminology.

Business Equipment Manufacturers Association (BEMA)

This organization of American producers of all kinds of office equipment acts as sponsor of ASA Sectional Committee X-3 on computers and information processing, and thereby furnishes some of the indispensable office support for X-3.4 and the other subcommittees of X-3. More importantly, some of the key members and staff of X-3 are BEMA staff members and play a vital role in keeping manufacturers interested in standardization.

European Computer Manufacturers' Association (ECMA)

Unlike BEMA, ECMA activities are almost exclusively concentrated on development and standardization in the computer field. Among its technical committees, TC 2 (General Programming Languages), TC 5 (ALGOL) and TC 6 (COBOL) work in the area of computer languages, ISO (see below) provides a common meeting ground between the ECMA and ASA committees.

International Standards Organization (ISO)

For almost 20 years ISO has served as the focal point for standardization on an international scale for a wide variety of goods and services. Forty-five countries are participating in its work. Among its numerous technical committees is one, TC-97, with the title "Computers and Information Processing". Its scope corresponds closely to that of ASA Sectional Committee X-3 with the same title. Like X-3, ISO TC-97 has set up a number of subcommittees, among which SC 5 (Programming Languages) corresponds to ASA X-3.4.

In few areas is the need for international validity of standards more thoroughly recognized than in computer languages. From the start X-3.4 and TC 97-SC 5 have been keenly aware of the overriding need for the closest cooperation in, and coordination of, all their activities. This has been achieved in several ways. First of all, many individuals hold positions in both organizations, so that there is mutual awareness of activities and motives in more detail than could be accomplished by merely exchanging documents. Secondly, ISO frequently relies on the adherent national groups for its technical work, and the United States, because of their dominant position in the worldwide computer market, have carried a large share of these responsibilities. Finally, each of the two organizations has shown complete willingness to take the activities of the other into account in planning its own work. Thus, as has been mentioned, the international standardization of FORTRAN and COBOL has been deferred pending action on the national level in the United States, while in the matter of ALGOL standardization X-3.4 has concentrated on cooperating in the ISO effort in the expectation that its results may later be adopted as American standards.

Outlook

In conclusion, let me point to the two main difficulties which stand in our way and prevent us from progressing as fast as we should like to. One of these, which I have mentioned earlier, is the unusual complexity of the subject, both economic and technical; the multiplicity of interests and effects, the difficulty of describing programming languages and their associate processors, of testing them and making sure of their completeness, consistency and compatibility. Another is simply the lack of manpower. X-3.4 has an extremely capable and enthusiastic group of committee members, but the really active ones are few in number and they are volunteers who take time out from their regular duties to work on standardization of programming languages. Because there are so few of them almost everyone is carrying multiple assignments; and these same individuals also serve in the other organizations discussed in Section 4. We need more help.

A third difficulty has been a lack of funds for travel for participation in committee meetings. This has forced us to look for participants principally in a small geographic area, stretching from New York to Washington, D. C., and it has prevented many capable and interested persons, especially in the academic world, from joining altogether. In both ways it has aggravated the already critical lack of manpower.

By detailing these difficulties, I do not mean to say that I am discouraged. On the contrary, I am confident that despite delays we shall soon reach our objective of an adequate set of standard computer programming languages.

I am grateful to my colleagues in X-3.4 for their advice and help in the preparation of this material. In particular, H. Bromberg, assisted by D. A. Goldstein, supplied the paragraphs dealing with X-3.4.4 and COBOL; P. Z. Ingerman and J. N. Merner checked and corrected the sections on X-3.4.2 and ALGOL, W. P. Heising and Jean Sammet, those on X-3.4.3 and FORTRAN. Furthermore, much of the material presented here is based on earlier reports prepared by S. Gorn and R. E. Utman.