

go unrecognised, and focusing attention onto lost opportunities.

With the esoteric content of programming eliminated, a much broader section of the population will become potential computer users. Not only would more people be capable of using a computer, but those who had previously considered it frustrating and time-consuming might now find it worthwhile. For example, with the advent of scientific management, managers could well use the computer for small simulations or analyses of local problems requiring immediate solution. In general, professional people can expect the routine tasks to be performed by computer, leaving them more time to concentrate on the fundamental aspects of their work. Apart from the sociological desirability of this trend, it should bring tremendous indirect advantages in the form of improved industrial efficiency, better designs, etc.

An area where direct cost savings would occur is in training. Accepting that less involved high level languages, such as Atlas Autocode, are more easily assimilated by new users, one would expect that training periods could either be shortened or a better training given in the same time. This would attract those potential users, such as senior managers, who have only limited amounts of spare time in which to learn to program. Unfortunately, at present, when a new computer system is installed or when programmers change jobs, the method of communicating with the computer has to be re-learned. Even with a widely used language such as ALGOL 60, different installations have different input/output procedures.

Here again, the simpler the language and operating system, the shorter the disruption of normal working after a changeover. This situation would be avoided by the use of standard languages and systems. One reason for not standardising is that some languages are more efficient than others for particular applications, but a penalty is paid in the enormous duplication of effort carried out. If a single computer language were introduced, it would still need to be translated into several other languages for use by other countries, but problems of interchangeability of programs and personnel would be reduced to a minimum.

Considerable thought and effort has been aimed at improving compiler efficiency. However, if one compilation error has been made by the programmer a corrected program will have to be run, doubling the computing time. Thus, any fractional advantages gained by a fast compiler are swamped by the wasted computing time caused by human error. The authors maintain that a considerable quantity of compiler efficiency could be sacrificed in order to accommo-

date language changes orientated towards reducing human errors.

Similarly, advantages of reducing the running speed of compiled programs are recognised, but less apparent is the time wasted through incorrect data or logical errors. It is clear to see how data can be mispunched, particularly in FORTRAN where each character has generally to be in a specific card column, but not so obvious that some logical errors can also be induced by poor language design. Admittedly logical mistakes are inherent in any human task, but some of the language improvements suggested above show how these might be reduced.

From the authors' experience at several establishments programs often fail because of computer operator mistakes. Such mistakes include feeding paper tapes in backwards; omitting to replace punched cards rejected by the card reader; failing to call up special procedures for graph-plotting or visual display; and failing to switch on required output devices. Some of these pitfalls could have been eliminated by better design of the system, giving direct cost savings. Introducing new improved systems is not necessarily advantageous however because adapting from an old system to a new one appears to require as much re-training and induce as many programming errors as changing from one language to another.

### Conclusion

When an overall view is taken of the effectiveness of present computing systems, serious deficiencies are seen in the functioning of 'high level' languages. With the expected increase in the number of occasional computer users, there will be a corresponding increase in the costs expended on training people to program and in the computer time wasted on program development. This emphasises the need for a computer language which is easy to understand, easy to remember and free from error prone conventions. Such a language is unlikely to be designed by computer personnel who have had no training in human factor techniques, and who are unconsciously ingrained with computer jargon that is incomprehensible to the majority.

Let us hope that those commissioned to write new languages, or even subsets of the old, will be sufficiently aware of the needs of the programmer to place him in his important position as the ultimate consumer, rather than to exploit his weaknesses with the tedious and, in many cases, unnecessary conventions inherent in existing languages and implementations.

---

## Book review

Annual Review in *Automatic Programming*. Vol. 6, part 4. 'Joss-II: Design Philosophy', by J. W. Smith, 1970; pp. 183-256. (Pergamon Press Ltd., 30s = £1.50)

Papers in which system designers make an honest attempt to review, explain and, where necessary, criticise, the decisions that they made in designing a software system are all too rare (though similar papers about hardware systems are virtually non-existent). This paper on the design philosophy of JOSS-II—and in particular the sections on list structures and list processing, conditional expressions and storage management, and the 'reprise'—is very worthwhile on this account. However, other sections of the paper describing the details of the JOSS-II

language are less successful. Much space is taken up by a rather discursive account of the rules of the language. A briefer, more formal description of the language, with more discussion of the alternatives which had been considered and discarded would have been preferable.

Even more regrettable is the lack of any attempt to assess the merits and demerits of the language, relative to other general purpose languages both conversational and conventional.

A paper such as this should not have had a 22-item bibliography in which nothing other than papers and reports relating to JOSS appears.

B. RANDELL (Newcastle upon Tyne)