WEXELBLAT84

Wexealblat,R.L.;
Nth Generation Languages;
Datamation (September 1984) pp. 111-117.

In this article the author identifies several generations of computer languages:
 machine or assembly languages represent the first generation, FORTRAN the second generation, while languages such a s Ada and Modula-2 represent the third generation.

Third generation languages are typified by features such as data abstraction and control of asynchronous processes -- features which are unavailable or difficult to implement in older languages. The fourth generation is still in evolution, however the languages of that generation will be rule-based rather than control-based as are languages such a PL/1 and Ada. The statements in the rule-based rather than control-based consist of conditions and actions. If more than one condition is true, then the consequent actions are executed in parallel or in random order.

The author goes on to make the following observation: there does not seem to be any great impetus for programmers knowing programming languages of second generation to learn and use third generation programming languages, and for programmers in third generation languages to switch to fourth generation languages. The reluctance to switch languages has its roots in the natural inertia on the part of programmers and managers to any changes thus making virtues of new languages almost irrelevant. This reluctance is further entrenched by a perception that newer languages have many of the problems present in earlier languages. New languages are difficult to learn, remember, and use. They have inadequate documentation and are still error prone. The conclusion that follows is hardly surprising: common lanugages of the year 2000 will be called FORTRAN-2000 and COBOL-99.

The author believes that we should abandon the idea that programming is the onl;y way to get the computer to do something for us. The way out of the trap of more and more programming is through creation of expert systems. This suggestion presented is that of combining and expert system for problem solution with an expert system for reuse of stored solutions. Whenever such a computer comes up against a problem soulution components of which are not stored, people will enter new solution components into the reusable component store.

The contemporary programming should be heading in the direction where programming is not just a task of specifying correct algorithms. The author says that we should break from the conventional mold of programming even though he does not himself quite know how to do that.

The final part of the article deals with the fourth generation of languages (rule-based) and some of the difficulties inherent in their implementation.

Whether or not one agrees with R.L.Wexelblat's basic premise that the computer can ever solve problems merely by having the proper level of explanation, it can be unequivocally stated that he raises some interesting questions in this article which are thought provoking.