WULF80

Wulf, William A.; Trends in the Design and Implementation of Programming Languages.; IEEE Computer 13, 12 (December 1980).

In this paper Wulf presents a subjective survey of the major areas of research in the area of programming languages. He begins by outlining some points about the field in general.

Wulf notes that research in programming languages has yet to make a large impact on the programming problem, as most programming is still done in old languages. Despite this shortcoming, he feels that language development must continue. He believes that programming languages are "the primary notational vehicle" of computer science, and most advances in the rest of computer science eventually find expression in programs. Thus, developments in programming are tightly bound to developments in other areas of computer science. Programming languages lead other fields by posing new problems such as formal proof techniques, and at the same time trail, by requiring new language constructs to handle new developments such as array architectures.

The paper then proceeds to divide current language research into four main areas. The first is application-specific languages. Many current languages of this type are misdirected as they add needed generality to a specialized language. Wulf argues that the proper method is begin with a general language and then add specific features.

The second area is very high level languages. Both their strength and weakness lie in their emphasis on "what" rather than "how". The strength lies in the ability of humans to better work with "what", but their weakness lies in the limitations of translation into "how".

The largest focus of current research is data-abstraction languages. Wulf maintains that the type definition feature of these languages enables the programmer to elevate himself to higher level of abstraction. He notes three main trends in these languages : abstraction, modularization, and verification. He feels that without a significant breakthrough in the tractability of formal proofs, that the emphasis on verification will have been largely misplaced. Finally, feature specific research is identified as the fourth main area of investigation. Features mentioned include : the goto, scope rules, iteration, concurrency, pointers, exceptions, and protection.

The paper divides the field of implementation into three main areas : compilation algorithms, automation, and run-time support. Research in compilation algorithms has reduced most compiler construction to an engineering problem, but great difficulties remain in very high level language translation.

Automation of compilation has been successful for limited parts of the compilation process. The paper targets the remaining areas for new research in the light of the recent proliferation of different processors. In the area of run-time support the issues of efficiency, portability, and diagnosis are briefly touched on.

Wulf believes that all this research is closely tied in with areas such as methodology, verification, style and hardware architecture. He concludes by offering four observations. 1) Programming is hard. 2) Future improvements will manifest themselves as solutions to more complex problems rather than cheaper solutions. 3) Programming languages have a very long cycle time, from invention through to acceptance, and 4) that language design cannot be isolated from the rest of computer science.

This paper presents a good overview of the setting of language design. In particular Wulf provides enough perspective to show that language design has a long way to go before it has a major impact on how the vast majority of code is written, and that to be effective language design must encompass concerns and concepts from a large part of computer science.