SUPL (SYNTACTIC UNIVERSAL PROGRAMMING LANGUAGE) — A NEW DIMENSION OF SOFTWARE DESIGN

C. Muses, R.C.M.M., Santa Barbara, Ca., (U.S.A.)

1. Among the important parameters of program success fall:

(1) Proofs of correctness (confirmed debugging);

(2) Precision of instruction (which includes minimization of unnecessary steps, since every extra step increses probability of processing errors); and

(3) Optimal command-formatting (i.e. best language).

Often (2) will result from the proper application of (1), so these may to advantage often be considered together.

2. It is with item (3) that this paper will be principally concerned. Today there are a plethore of various programming languages — so much so that many machines in use can be programmed in one or a mixture of two such languages, but not in a third. The reason for this is inflexibility of software, reflecting in turn a too inflexible hardware.

3. Thus today we have a virtual level of programming and software languages, whereas the list of intelligent and useful things to say is comparatively unique. To eliminate this needless duplication and inflexibility, we went through existing languages and listed, non overlappingly, the uniquely useful commands of each, including-assembly language. Then we added some others, not yet in an existing language, and we arrived at a surprisingly flexible, powerful and reliable language:

SUPL ("pronounce "supple"): Syntactic Universal Programming Language. Moreover, SUPL at once suggests how current computers may, with inexpresive changes, and in some cases no change at all, be easily adapted to SUPL processing with consequent tremendous gains in clarity, worldwide communicability, powerful flexibility, and greatly increased efficiency and reliability. A large bonus is the break through provided by SUPL in the hitherto difficult area of natural language translation.