Letter to the editor

Formal representations for EMS – job control language

Sir, with reference to 'Towards a formal representation of EMS' by Richard Miller and Jacques Vallee,[1] the authors' attempts to build up an EMS background, using BNF algebra and coding sheets, seem promising. There is, however, a point that I would like to make.

To quote from by own contribution to *Telecommunications Policy*,[2] entitled 'The power of local information':

> . . . in the longer term, free-text conferencing will probably be interpreted in a partly automated way. Certain conference sentences, more or less freely intermixed with other sentences, may be 'understood' by the computer system, leading to specific formal actions. A mixture of syntactic and semantic grammar might be used. Integrated parts of the dialogues could be treated as 'comments', while other parts could be interpreted as instructions. It may also be that these two forms in certain cases could be mixed in a dynamic way: comments for action, so to speak, and instructions that could be ignored. Such dynamics would then have to be controlled from higher system levels.

Today, one has the impression from discussions on software development that such dynamics, between executable and non-executable statements in a text, will play a role in future systems. The criteria for selecting the relevant modes for different text parts might be directly application-dependent. Seen from application one, the entering text *A* is a comment, while the follow-up *B* contains executable instructions. Application two, however, regards the same starting text *A* as a formal procedure that is going to be executed after certain actual data are provided. *B*, then, could be ignored in this context.

If there is no *a priori* difference in a text between non-executable strings (comments) and executable instructions (or program procedures), then the formal text background should be of a general type.

There is already quite thorough experience available concerning job control languages (JCL) in computer-operating systems. These have lately been developed to contain fairly complete grammars. Here is a tool for handling text and text procedures that ought to be looked at while new text functions and control strategies are created for formal representation of EMS.

Internationally, coordination activities concerning common command languages (CCL) are centred around the work in IFIP WG 2.27. Recently, North Holland has published proceedings from this group's meeting held in West Germany, September 1979.

In the USA, both ANSI and CODASYL have active CCL committees. It appears that the latter of these two also has arrived at detailed proposals.

It is natural that the CCL activities need time for further development. The need for increased command language intelligence comes to mind. Computer system actions should be able to be taken without the need for great detail in control commands. The human interface should be strongly borne in mind, as always.

Furthermore, computer networks, including personal computers, raise new demands on CCL.

Generally, it is likely that there are interesting resemblances between formal languages on different levels of abstraction. Analogies have always been fruitful in natural science. I suggest that a careful look is taken at existing and planned job control languages in the work of defining formal representations for EMS.

Tomas Ohlin,
Commission on New Information Technology,
Stockholm, Sweden

[1] Richard H. Miller and Jacques F. Vallee, 'Towards a formal representation of EMS', *Telecommunications Policy*, Vol 4, No 2, June 1980, pp 79-95.
[2] Tomas Ohlin, 'The power of local information', *Telecommunications Policy*, Vol 2, No 3, September 1978, pp 234-243.