Session 10: PROGRAMMING

SUMMATION BY CHAIRMAN

WARE: As many of you know, I am not a programmer myself although I am conversant with the language of programmers and, therefore, my summary comments are from the point of view of a disinterested non-practioner of the programming arts. Furthermore, I am not familiar with most of COMIT and MIMIC and the other systems which have been proposed, and so my remarks are necessarily based on what I have heard presented this afternoon. An observation in passing--we as human beings converse in what we choose to call the English language. It turns out that the subset of the English language which is convenient for stating problems to the programmer is a lot different from the subset of the English language which is convenient for stating problems to the engineer. I am therefore not particularly surprised if COBOL does not turn out to be convenient for MT. In fact, I think I would be a little suspicious if it did turn out to be convenient for MT. It is not clear to me that our insight into the structure of pseudo-languages as well as insight into the problems themselves is sufficiently deep at this stage that we ought seriously to think about constructing languages which are universal in some sense. I, for one, am not particularly partial to the notion that MT should have a big hand in defining how COBOL might go. I agree it might be convenient, but I am not convinced that it is healthy for the research in the field. There have been a few comments that have been made this afternoon that caught my ear in passing. Perhaps a recitation of these will serve as a summary. Simulation of one machine by another machine is certainly appropriate when we are thinking of research, and especially research in an area which is difficult, and in which there is virtually an absence of an underlying theory. But when we pass from research into what we might call production, then the simulation of one machine by another, or the necessity to consume excessive amounts of machine time, is at an economic disadvantage and we had better do something about it. There seems to be an implication that you machine designers may have to provide storage devices that are bigger than 32, 000 words, or that you may have to provide storage hierarchies which are sufficiently more efficient than the present ones. It seems to me that we are beginning to see people thinking about measures of relatedness between words where the measures of relatedness is not simply the binary digit 0 or 1. It would appear that we are beginning to see the influx of probabilistic couplings between words or groups of words or pairs of words, and perhaps this is one of the most exciting and interesting developments that will come in this field in a year or two. My impression from hearing both the presented papers and the panel is this: the programming art as practiced in MT is not different from the programming art practiced anywhere else. The state of development of compilers of sophisticated languages is no better here and it does not appear to be significantly poorer either. So it looks to me as though the programmer is keeping abreast of this new kind of problem as well as he is keeping abreast of new scientific work generally.