Weggeschickt am 12.6.87

KNOWLEDGE SPECIFICATION AND INSTRUCTIONS FOR A VISUAL COMPUTER LANGUAGE

CLAUS MÖBUS

University of Oldenburg, D-2900 Oldenburg, W-Germany

a) We propose the use of logic programs designing intelligent tutoring systems. With the help of these programs we specify curriculum, design interface and derive instructions.

b) Today there is considerable agreement about the strategic aspects of designing an intelligent tutoring system (ANDERSON, 1987; ANDERSON, BOYLE, FARRELL & REISER, 1984; DEDE, 1986; WOOLF & McDONALD, 1984). However, there is less conformity about tactical aspects, concerning the structure of knowledge bases and the quality of instructions and interfaces. It seems to us that concentration on developing latent components (e.g. flexible student-models and error-explanation algorithms) has led to a certain neglect of manifest components (interface and instructions). A good example of this argument are some shortcomings in ANDERSON's well known LISP-Tutor (ANDERSON, 1987).

In the talk we want to indicate first steps towards the realization of our concept. The domain of discourse is functional programming with a graphical computer language. The proposed language is adaptive: it possesses complete visibility of all computational steps, if the user is a novice. During development expertise, the language becomes more and more abstract. The programming environment is going to be implemented on an INTERLISP/LOOPS -workstation.

Because our programming language existed only as on thought level in informal texts and drawings (BAUER & GOOS, 1982), we decided to make a knowledge-specification. We specified with rule-sets the necessary minimal semantic and syntactic knowledge a student has to master before he is able to follow planning instructions successfully. The rules were written in PROLOG so that they could be used as a runnable specification (DAVIS, 1982).

From the PROLOG facts, describing static characteristics of our functional programs, we derived graphical elements which are the building-bricks of our graphical programming language. From the PROLOG rules, describing the control and dataflow of the graphical programs, we derived combined natural language and pictorial instructions.

Furthermore, it will be shown in the talk how to model the knowledge-acquisition process of a student with Horn-clause-rules. The development is described as a transition-path through a state-space. Each state is represented by a rule-set containing the knowledge of the student.

ANDERSON, J., Production Systems, Learning, and Tutoring, in: D.KLAHR, P.LANGLEY & R. NECHES (eds), Prduction System Models of Learning and Development, Cambridge, Mass.: MIT Press, 1987, 437 - 458

ANDERSON, J.R., BOYLE, C.F., FARRELL, R. & REISER, B.J., Cognitive Principles in the Design of Computer Tutors, Advanced Computer Tutoring Project, Carnegie-Mellon University, 1984

DAVIS, R.E., Runnable Specification as a Design Tool, in: K.L.CLARK & S.A.TÄRNLUND (eds.), Logic Programming, New York: Academic Press, 1982, 141 - 149

DEDE, C., A Review and Synthesis of Recent Research in Intelligent Computer-Assisted Instruction, International Journal of Man-Machine Studies, 1986, 24, 329 - 353

WOOLF, B. & McDONALD, D.D., Building a Computer Tutor: Design Issues, Computer, 1984, Vol 17, No 9, 61 - 73

d) This research was sponsored by the Deutsche Forschungsgemeinschaft (German Science Foundation) in the SPP Wissenspsychologie (Psychology of Knowledge) under Contract No. Mo 292/3