Volume 3, Number 3, 1988

Construction Kits and Design Environments: Steps Toward Human Problem-Domain Communication

Gerhard Fischer and Andreas C. Lemke

Our goal is to build cooperative computer systems to augment human intelligence. In these systems the communication between the user and the computer plays a crucial role. To provide the user with the appropriate level of control and a better understanding, we have to replace human-computer communication with human problem-domain communication, which allows users to concentrate on the problems of their domain and to ignore the fact that they are using a computer tool.

Construction kits and design environments are tools that represent steps towards human problem-domain communication. A construction kit is a set of building blocks that models a problem domain. The building blocks define a design space (the set of all possible designs that can be created by combining these blocks). Design environments go beyond construction kits in that they bring to bear general knowledge about design (e.g., which meaningful artifacts can be constructed, how and which blocks can be combined with each other that is useful for the designer. Prototypical examples of these systems (especially in the area of user interface design) are described in detail and the feasibility of this approach is evaluated.

A Keystroke Analysis of Learning and Transfer in Text Editing

Mark K. Singley and John R. Anderson

Two experiments studied the acquisition and transfer of text-editing skill. The first experiment,originally reported in Singley & Anderson (1985) but reanalyzed in greater detail here, found nearly total transfer between two similar line editors and partial transfer from the line editors to a screen editor. Analyses of the keystroke data revealed that the majority of the improvement during both learning and transfer was concentrated in the planning components of the skill. The second experiment found little evidence for negative transfer between a pair of screen editors designed for maximal interference using a classic interference paradigm. The few instances of negative transfer observed were better characterized as the positive transfer of non-optimal methods rather than instances of true procedural interference. These results support an identical elements model of transfer based on a production system representation of cognitive skill. The relative magnitudes of transfer observed were consistent with detailed measures of production system overlap. In addition, localized transfer sites were hypothesized and identified through a series of microanalyses. Finally, specific transfer predictions based on the differential practice of general and specific components were tested and confirmed.

Animation Using Temporal Constraints: An Overview of the Animus System

Robert A. Duisberg

Algorithm animation has a growing role in computer aided algorithm design, documentation and debugging, since interactive graphics is a richer channel than text for communication. Most animation is currently done laboriously by hand, and it often has the character of canned demonstrations with restricted user interaction. Animus is a system that allows easy construction of an animation with minimal concern for lower level graphics programming. Constraints are used to describe the appearance and structure of a picture as well as how those pictures evolve in time. The implementation and support of temporal constraints is a substantive extension to previous constraint languages which had only allowed specification of static state. Use of the Animus system is demonstrated in the creation of animations of dynamic mechanical and electrical circuit simulations, sorting algorithms, problems in operating systems, and geometric curve drawing algorithms.