Abstract

Visual Programming

Jeffrey V. Nickerson

Advisor: Professor Edmond Schonberg

While computer science textbooks and classroom lectures are filled with diagrams, and much of our design activity as programmers takes place on whiteboards, we write our programs as text. Proponents of visual programming suggest that we should take advantage of graphic user interface technology and draw rather than write our programs. This dissertation examines the extent to which this is possible, addressing the question of how graphic representation can best be used in the process of programming.

The use of diagrams in the field of computer science is thoroughly surveyed, and some underlying principles identified. The visual conventions of Adjoinment, Linking, and Enclosure are defined and illustrated. Three languages are developed - a simple programming language that encompasses shell commands, a visual version of APL, and a visual front end for Mathematica. The visual version of APL is notable in that it presents both a program and instances of data undergoing transformation as part of one unified diagram.

Building on the work of R. J. A. Buhr, new visual systems designing conventions are created to handle the intricacies of facilities in the Ada 9X language. Asynchronous transfers of control, requeueing, and generic formal parameters are addressed. The asynchronous transfer of control convention is suitable for CASE representations of the language construct, and can be easily animated.

Some existing software metrics are modified for use in analyzing diagrams, and two new metrics are proposed: graphic token count and diagram class complexity. A graphic design measure, data density, is transformed into a computer science measure, token density. Using these metrics, graphic representations can be compared to each other and to textual representations. From this, a strong set of conclusions are drawn about the relative strengths of graphic and textual representation, as well as the limits and possibilities of graphic representation in programming.