KEEPING TRACK: COORDINATING MULTIPLE REPRESENTATIONS IN PROGRAMMING

Pablo Romero, Benedict du Boulay and Rudi Lutz IDEAs Laboratory University of Sussex

A computer program is a complex object with both static and dynamic properties that can be viewed from a number of different perspectives. These perspectives include the functional relationship between different parts of the code and what they do, flow of control information about the sequence of events when the code executes, the nature and evolution of the data structures built dynamically during execution and so on.

Modern software development environments exploit their graphical interfaces to provide varied notations and modalities to visualise these and other perspectives. One dimension of this variability is in terms of modality, e.g. from mostly textual to mostly graphical.

An effective programmer, using a software development environment well, is able to coordinate the various representations available in a way that suits both the particular constraints of the kind of programming task in hand (module design or debugging, say) and the context in which it is being undertaken, as well as any personal preferences for, or expertise with, particular styles of representation.

This paper compares how programmers of varying expertise coordinate four simultaneously available but different representations of a Java program they are debugging. These representations are the code of the program, its flow of control, its data structures and its printed output. The paper explores which representations get used and when, the pattern of switches between representations, the effects of individual differences in preference for particular modalities or in expertise in switching between modalities, and the relationship of these factors to debugging performance. The paper describes a novel method of tracking the programmer's focus of attention.