

Abstraction levels in editing programs

John C G Sturdy

University of Limerick
john.sturdy@ul.ie

1 Introduction

Traditionally, text editors for program text have operated largely at the level of lines and characters, and their users have made their changes largely character by character, with “cut and paste” and “find and replace” as their only more powerful actions.

Some editor users may think of changes to source code in those terms, too, but it is also possible that some, perhaps the more experienced programmers, think at a higher level of abstraction, such as syntax trees, and would be able to work more efficiently with tools that match the way they think.

Some such tools have now begun to appear, typically described as “refactoring tools”; some of them are integrated into editors and Integrated Development Environments (IDEs) such as Eclipse, NetBeans, IntelliJ, and Visual Studio; and some are separate command-line programs.

It may also be the case that the provision of such tools will help to train or educate programmers to think of their work in a more abstract way; this effect may appear naturally as experienced programmers go about their work, or could be done deliberately in the training of novice programmers.

This research sets out to explore some editing tools that work at higher levels of abstraction than lines and characters, investigating first their design, and then moving on to empirical research to find their usefulness and effects.

2 Survey

Card, Moran and Newell[1] compare navigation using traditional cursor keys, keys that produce movements in terms of the text units (paragraph, line, word), joystick and mouse. They found the mouse was fastest, followed by joystick, then character cursor keys, with the text movement keys being the slowest. Their experiment was for some word processing tasks; one of the tools presented in the present research is equivalent to the one they found slowest, but in the context of editing source code, which, being more strongly structured, may give more of an advantage to movements based on the text contents. Also, it is possible that using the text contents to guide the movements and editing operations may reduce the number of small corrections at the end of a sequence of keystrokes. Despite the advantages claimed for the mouse by usability experts and by novice users, many experienced computer users, however, largely prefer the keyboard over the mouse

