Part 3

Expert Programming Skills and Job Aids

After having examined programming language features, their learning by beginners, their implications on the structuring of programming knowledge, and their use in education (Part 2), this part is devoted to programming at an expert level. The perspective is wider, less dependent on programming languages than that of the previous part, and relevant to other design activities.

As shown in Part 1 of this volume (Chapter 1.3), programming combines several features which together form expert skills. Two basic skills have received much attention at an expert level: program understanding and designing. These are basic in that they are essential components of other skills. For example, program debugging is closely linked to program understanding: experts elaborate a hierarchical representation of the program control structure and function before beginning to detect or correct errors (Vessey, 1985, 1989). Programmers form a mental reconstruction of what a 'correct' program is, to compare it to the actual program (Michard, 1975); hence the close link between program debugging and program designing.

The three chapters in Part 3 exemplify the need to decompose a complex programming activity in order to gain a better understanding of its components. This carries the concomitant risk of transforming these components in the simplified situations under study. However, Chapter 3.3 and Part 4 address the question of 'programming in the large' in more complex and everyday work situations. The chapters in Part 3 shed light on complementary features of programming, rather than taking exclusively theoretical positions.

The first two chapters present two kinds of approaches to program understanding, and reflect the fundamental duality of cognition calling for use of declarative and procedural knowledge at the same time (see Chapter 1.4). Program-understanding is studied, for example using verbal reports on expectations during program reading, or in the context of debugging or recall tasks, etc. Some data analysis techniques are more directed to identifying expert declarative knowledge bases, especially program plans and control structures used in program understanding (Détienne, Chapter 3.1). Other techniques are needed to investigate understanding strategies more directly, and to examine the respective role of expert knowledge bases and expert strategies in the program understanding expertise (Gilmore, Chapter 3.2). Implications in terms of programming teaching touch on the importance of teaching both kinds of knowledge.

Aside from these investigations into program understanding, studies have been devoted to expert design strategies. The last chapter of Part 3 reviews these studies, presents dimensions used to classify strategies and some recommendations for developing job aids (Visser and Hoc, Chapter 3.3). In this chapter, programming is considered as a particular case of a design activity and the framework and recommendations go beyond the sole computer programming.

References

Michard, A. (1975). Analyse du travail de diagnostic d'erreurs logiques dans un programme FORTRAN. Le Chesnay (F.), INRIA, Research Report no. C07602R48.

Vessey, I. (1985). Expertise in debugging computer programs: a process analysis. *International Journal of Man-Machine Studies*, **23**, 459-494.

Vessey, I. (1989). Toward a theory of computer program bugs: an empirical test. *International Journal of Man-Machine Studies*, **30**, 23-46.