Chapter 25
Experiences in Usability Evaluation of Educational Programming Tools

J. Ángel Velázquez-Iturbide
Universidad Rey Juan Carlos, Spain

Antonio Pérez-Carrasco
Universidad Rey Juan Carlos, Spain

Ouafae Debdi
Universidad Rey Juan Carlos, Spain

ABSTRACT

This chapter advocates for an approach to constructing educational tools that consists in designing small systems aimed at achieving clear educational goals and evaluating them in actual teaching situations. The authors addressed this approach with a number of small systems. In this chapter, they describe their experience in the development, use, and evaluation of two educational systems: SRec and GreedEx. The former is a highly interactive program animation system of recursion, and the latter is an interactive assistant aimed at learning the role of selection functions in greedy algorithms by means of experimentation. The evaluations allowed the authors to identify faults and weaknesses of the systems, and these results were used to enhance the systems. Moreover, their approach has yielded very high values with respect to effectiveness and student satisfaction.

1. INTRODUCTION

Currently, universities intensively use computer systems in order to improve, simplify, and even accelerate teaching and learning tasks. These systems are technologically sophisticated, and this trend seems to continue in the future. Unfortunately, many educational systems suffer from one of these drawbacks: either they are not targeted to educational goals, or they are not tuned to actual use, in real-world teaching situations.

An approach to deal with this problem consists in designing small systems aimed at achieving clear educational goals and evaluating them in actual teaching situations. Having clear educational goals as the main part of their specification results in very

DOI: 10.4018/978-1-4666-7363-2.ch025

effective tools for the tasks they were intended. Furthermore, taking into consideration practical issues (such as installation, internationalization, etc.) removes obstacles for their adoption by other teachers. Finally, the evaluation of these systems allows tuning them for effective use, but it also allows discovering incomplete support to certain tasks or additional features that were not envisioned by their developers.

Our group has addressed this approach with a number of small systems. This chapter illustrates the issues pointed out above, such as designing educational software based on educational goals, consideration of real-use issues, and evaluation of the tools. As driving examples, we show our experience in the development, use, and evaluation of two educational systems: SRec (Velázquez-Iturbide, Pérez-Carrasco, & Urquiza-Fuentes, 2008) and GreedEx (Velázquez-Iturbide & Pérez-Carrasco, 2009). The former is a highly interactive program animation system of recursion and the latter is an interactive assistant aimed at learning the role of selection functions in greedy algorithms by means of experimentation.

The chapter is structured as follows. In the second section, we introduce some concepts, which are necessary to understand the chapter, namely programming and algorithms, software visualization, learning goals, systems for programming education, and usability. In the third and fourth sections, we describe the respective features of the SRec and GreedEx systems, and show the results of evaluating them for usability. The fifth section contains a discussion of the results, presenting them to other developers as lessons learnt. Finally, we outline our conclusions and identify lines of future research.

2. BACKGROUND

In this section, we introduce some issues, which are preliminary with respect to the rest of the chapter. Firstly, some concepts of programming and algorithms are briefly introduced, such as recursion and greedy algorithms. Secondly, the main concepts on visualization and animation are given. Thirdly, Bloom's taxonomy is outlined as a framework to state learning goals. Fourthly, we review the myriad of existing systems for programming education, with an emphasis on their explicit learning goals. Finally, the usability methods we have used will be identified and placed in the context of the usability methods currently available, mainly expert evaluations, questionnaires, and observations.

2.1. Programming and Algorithms

Programming and algorithms are core topics in Computer Science (CS) education (ACM, 2008). We only introduce here some concepts, which are relevant for the rest of the chapter, namely recursion and greedy algorithms.

Recursion is a linguistic mechanism consisting in defining something in terms of itself. Recursion is strongly discouraged for definitions in general, and dictionaries try to avoid it. However, in mathematics and computer programming, recursion is a valuable tool for definitions and for problem solving. Some recursive definitions are surprisingly clear, being the Fibonacci series a paradigmatic example.

Recursion is one of the most difficult concepts in procedural programming and therefore it has received much attention from CS educators. Many conceptual models have been created in order to facilitate student learning. Most conceptual models have a strong visual component, e.g. the copies model or recursion trees. See the review by Pérez-Carrasco and Velázquez-Iturbide (2012).

Algorithms are detailed descriptions of solutions to problems stated in terms of input data-output data. Algorithms have several properties that can be modeled mathematically, e.g. correctness and efficiency. They can also be subject to experimentation, allowing the empirical study of their properties.