"Abstract Data Types as tools for Knowledge Representation and Problem Solving in a Prolog Environment"

Thesis for the Degree: DOCTOR of PHILOSOPHY

Submitted to the Scientific Council of The Weizmann Institute of Science, Rehovot, Israel

Development and Evaluation of A Computer Science Curriculum

Submitted by: Bruria Haberman

May 1998

Abstract

The work described in this thesis includes the development, implementation and evaluation of a computer science educational package aimed at introducing abstract data types (ADT) in the Prolog environment. This education package was developed as part of the "Computer Science – Logic Programming" course which is a third unit (out of five) of a new computer science curriculum, and as a part of an "Introduction to Artificial Intelligence and Expert Systems" course, which is an optional subject (out of four options) of the information technology curriculum.

The target population was composed of four groups of students that differed in the extent of their exposure to abstract data types and in the opportunity to use ADTs as tools for computer program development. The groups consisted of both novice and advanced students- those with expertise at basic level, advanced level and expert students.

The abstract data type concept was introduced to students according to a "flexible instructional model", taking into consideration the students' backgrounds and abilities, as well as course goal and pedagogical and other assumptions. The ADT concept was presented at three levels- (1) Specification and (2) Usage- to the entire student population and (3) the Implementation level- to advanced students only. All students learn how to use the "black box" that represents an ADT during the development of computer programs which represent knowledge or solve a specific problem. In addition students use ADTs to develop final projects (computer systems at various levels) as part of their studies.

The educational package developed here includes:

- Instructional materials concerning abstract data types for novice and advanced students;
- A teacher's guide with pedagogical instructions for teaching novice students;
- "Black boxes" that represent abstract data types implemented as Prolog programs.

The ADT instructional package was taught to Israeli students from 1991-1998 in academic and technological high schools. The teachers who taught the subject went through an appropriate in-service course.

An evaluation was conducted to develop the instructional package through stepwise refinement, considering the feedback from its actual implementation in the schools. The research concerned cognitive aspects of using ADTs as tools for knowledge representation and problem solving in a Prolog environment, as well as students' achievements and attitudes towards ADTs. The research was conducted at two levels: the product level – which considered computer programs (or computer systems) aimed at solving a specific problem, and the process level – which considered the program's development.

The main research goals were to determine the following:

- If and how students actually use ADTs to describe a problem?
- How are the ADTs used in terms of the product level and how is the product's quality influenced by their use?
- How ADTs are used in problem solving and knowledge representation?
- What is the student's perception of a "black box?"
- How does the extent of exposure to ADTs influence their use as tools in problem solving and knowledge representation?

The research target audience included stuff=dents that had previously studied abstract data types in a \prolog environment.

Using a combination of qualitative and quantitative methods, the work originated as a pilot ethnographic research on passive observations aimed at identifying possible methods of using ADTs. This pilot research was used as the basis of research concerning the product and development levels. In order to assess the development level, we followed teams and individuals while they developed computer projects. In order to asses the product level, we asked students to solve diagnostic questionnaires aimed at classifying their response in using ADTs. Students' solutions to problems were qualitatively and quantitatively analyzed. Students were interviewed and asked to both explain and justify their solutions. Student' final projects were analyzed as well. In addition, students were asked to express their attitudes towards ADTs by means of questionnaires and interviews.

The research findings indicated that the flexible instructional model for representing abstract data types that we developed enabled the teaching of ADT as a computer science concept and its use as a tool for problem solving and knowledge representation to various high school student populations. Furthermore, the findings indicated, that for many students, abstract data types played an important role as a project development organizer and contributed to a more structured and organized development. Using abstract data types made it possible for the students to concentrate on understanding and solving interesting and complex problems, creating simple computer programs using "black boxes", rather than dealing with

technical details related to the implementation level, this alleviated conceptual difficulties.

Specifically, the research tried to determine how students grasp and use the ADT concept. Indeed, most of the students adapted various methods of using abstract data types and "black boxes" for developing programs. Some students used methods from the instructional model; others invented methods of their own. Research findings have indicated that students use "black boxes" in various ways to develop computer programs. Although some used "black boxes" according to the instructional model, proving that they correctly grasped the ADT as a computer science concept, others used "black boxes" in ways that indicated that their concept of the ADT did not match its definition as defined in computer science.

In addition, the research results showed that a greater exposure to abstract data types doesn't necessarily lead to expertise in using them accordingly to the instructional model for problem solving and knowledge representation. It appears that students whose expertise in the subject at an advanced level showed irregular behavior while using the "black boxes". These students, as well as their colleagues, at first studied how to use the "black boxes" without knowing how they were implemented, and in the continuation of their studies they were intensively exposed to the implementation. Apparently that the exposure to the implementation of the "black boxes" temporarily diminished students' abilities to use them correctly. However, expert students who continued their studies showed and improvement in the style of using ADTs and demonstrated better performance compared to their colleagues.

The use of abstract data types by novice students contributed to the correctness of their developed programs. However we did not find a similar influence of ADT usage with advanced student projects.

Student' attitudes regarding abstract data types were positive and most of them agreed that it contributed to the development of computer programs.

An evaluation of the teaching unit showed that it was successfully implemented. The instructional model that was developed in this study enabled teaching abstract data types as a central computer science concept to high school students who successfully used them as a tool for knowledge representation and problem solving in a Prolog environment.