מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Thesis for the degree
Doctor of Philosophy

עבודת גמר (תזה) לתואר
דוקטור לפילוסופיה

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

מוגשת למועצה המדעית של
מכון ויצמן למדע
רחובות, ישראל

By
Giora Alexandron

מאת
גיורא אלכסנדרון

היבטים בלמידת שפת ה- LSC ותכנות מונחה-תרחישים
Issues in learning the language of LSC and scenario-based programming

Advisors:
Prof. David Harel
Dr. Michal Armoni

מנחים:
פרופ' דוד הראל
ד"ר מיכל ארמוני

July 2014

תמוז תשע"ד

Abstract

This thesis deals with computer science education, particularly with issues in learning the language of *live sequence charts* (LSC) [20] and scenario-based programming [38]. The focus of the research is on the cognitive aspects of the learning process. The main purpose of my work is to study the potential of LSC and scenario-based programming as a platform for teaching fundamental ideas of computer science, and to examine its learning process, in order to characterize factors that affect the learning and use of new programming languages.

Specifically, in the context of fundamental ideas of computer science (CS) [55], the study examines the effect of learning LSC on the ability to understand nondeterminism (ND) and abstraction. In the context of the factors that affect the learning and use of new programming languages, the thesis studies the effect of LSC on programmers' tendency to concentrate on usability or implementation aspects of the system they are developing, and on the effect of previous experience in programming, which is mainly procedural and object-oriented, on the learning of LSC.

As this study is the first to examine LSC in the context of computer science education, a bottom-up, *grounded theory*-oriented approach [30] is taken, using (mainly) qualitative research methods, and collecting data in varied research settings. This approach allows us to identify and characterize significant questions, and to suggest theoretical frameworks for addressing them. From these, more focused hypotheses that can guide further research can be derived.

Regarding the effect of learning LSC on internalizing ND, the findings show that high-school students can reach a significant understanding of the kind of ND that appears in LSC, when the concept is introduced in the context of a programming course on LSC and scenario-based programming. Regarding the effect of learning LSC on the use of abstraction, the findings show that when working with LSC programmers tend to think on a higher-level of abstraction and hold a less detailed mental model of the system they are developing, than when working with object-oriented programming (OOP) languages. A main conclusion from this part of the study is that LSC can be an effective platform for teaching ND and abstraction.

Regarding the effect on programmers' perception, the findings show that when working with LSC, programmers tend to hold a perspective that is more user-oriented, and give more attention to usability issues, than when working with OOP languages. This is

explained by a cognitive model, which describes how the kind of programming that LSC fosters allows programmers to spend more time in the problem domain.

Regarding the effect of previous programming experience, the findings shed light on several ways in which programming background that is mainly procedural and object-oriented can bias the learning of high-level programming languages. For example, this can lead students to design their programs in a way that limits the concurrency that is inherent in the language, so they can use known solution patterns, instead of developing new solution patterns that exploit the power of the new paradigm. The analysis, done through the prism of the *schema theory of learning* [4], characterizes some conceptual bugs that programmers can exhibit, and suggests some possible remedies. It also indicates limitations of this theory in explaining some aspects of the influence of previous experience on the learning and use of new programming concepts.