Understanding Concurrency: the Process and the Product

Thesis for the degree Doctor of Philosophy

By YIFAT BEN-DAVID KOLIKANT

February,2003

Submitted to the Science Council of the

Weizmann Institute of Science

Rehovot, Israel

ABSTRACT

My work integrated the development of a course in concurrent and distributed computation (CDC) for computer science (CS) high school students with an investigation of the dynamics of the knowledge evolution of students who studied this course. CDC has the potential to serve as an effective entry point for CS students into the practice of the academic CS world, which is the goal of CS studies in high schools, because it bridges between technology and the academic practice. On the one hand, it provides students with an explanation of computerized systems in the real world, and on the other hand, it deals with the non-deterministic nature of multi-entity systems, which require the students to acquire professional practice, thereby gaining the perspective of CS professionals.

The research focused on the students' knowledge growth concerning synchronization, which is a central topic of the course. The research questions were as follows:

- What are the typical pieces of knowledge students construct and append to their existing knowledge as they study the course?
- By what process does this knowledge evolve?
- How can the answers to the first two questions be utilized in order to improve the instruction of the course?

The research consisted of two phases: the preliminary research and the main phase of the research. The preliminary research was conducted in order to get a feel for the area. The students were found to possess highly non-viable pieces of knowledge regarding synchronization; the knowledge itself was found to be tangled and interconnected. During this phase, the observed difficulties of the students were categorized, and in addition, the students' attitudes to the course were explored and found to be positive due to the sense of the reality of the topics that they studied. Finally, this research provided the initial evidence on the evolution of students' understandings that

was reflected in their improved performance and the growth of their conceptions.

The main phase of the research focused on the topic of synchronization using semaphores. This phase involved (a) an exploration of the existing knowledge of the students at the beginning of the course, (b) an exploration of the students' knowledge at the end of the instruction of the topic with emphasis on incomplete pieces of knowledge, and (c) a reconstruction of the process of evolution of these pieces of knowledge during the course.

The data was collected almost exclusively from integral in-class activities, since any external interference would affect the process of learning. The investigation used a diversified set of tools to obtain various perspectives on the knowledge structures of the students. The research tools consisted of class assignments and tests that were collected from the entire research population. In addition, observations were made of one class during the entire period of the instruction of the topic. Additional data collected from this class included transcripts of videotaped laboratory problem-solving sessions, and interviews with several students. The process was repeated in the next year in another class. Two levels of inquiry were utilized: a macro-level view of an entire class for the general perspective and a micro-level view of individuals for deep inquiry of phenomena.

Although the students' formal CS education consisted of two introductory courses with no references to CDC, the students were found to possess informal prior knowledge that influenced their expectations, decisions and understanding of synchronization. Specifically, the students were found to be members in two relevant cultures. The first is of technology users—who are devoted to the production and manipulation of technology products and who work out problems at the interface level, by employing trial and error in order to get their product to "work" (Turkle, 1999). In contrast, CS education is oriented to the academic CS community, whose interest is in the abstraction, solution and proofs of algorithmic problems. Thus, CS formal education brings together old-timers in two different cultures that are interested in computations. This situation leads to a culture clash.

The second culture in which the students are members is the school culture. Participants in this culture play didactical-games and their goals are to win and survive these games (Broussaeu, 1997). This culture hosts the encounter between the technology users and the academic CS culture, and therefore crucial affects the students' dynamics of knowledge evolution. The students, old timers in the computer world, might judge the didactical activities offered in CS lessons as school obligations which are irrelevant to their practice, and accordingly would decrease their level of participation to mere survival.

The investigation of the students' preconceptions at the beginning of the course confirmed that the students are indeed members in these two cultures. The influence of the culture of technology users was reflected (1) in many students' belief that merely plugging in communication devices automatically resolves synchronization problems, (2) in the propensity to incorrectly use pre-defined programming structures, and (3) in the common propensity not the verify correctness thoroughly, but rather to rely on the results of a few executions. The propensity of many students to knowingly simplify tasks on the expense of the reasonableness of the solutions to synchronization problems demonstrates the influence of this culture.

It was found that these cultures influenced the students' judgments of productivity and therefore determined the shape of the knowledge structures constructed during the instruction. This insight explains the following findings: (1) students developed a technique to solve synchronization problems using problems previously learned at class as patterns, and (2) many students failed in assignments where no pattern could be used, but rather required them to explore the dynamics of programs execution (EDoE). Their failure was causes because they had not mastered the definitions of the structures that comprise the programs, such as semaphores, and showed no propensity to obtain this information. The analysis of the students' conceptual knowledge revealed that these students possessed the coarse-grained pieces of knowledge needed for EDoE, but that when fine-grained knowledge was needed, they (sometimes incorrectly) borrowed explanations from similar structures studied in the past. Thus the dynamics of knowledge evolution of these students is of a faulty extension of past productive work habits typical to technology users.

However, the investigation showed that many students underwent a process of enculturation, gradually crossing the boundaries from merely being technology users toward approaching computational situations as programmers. More specifically, it was shown that there is a continuous interplay between the cultural aspect of knowledge and the cognitive aspect of knowledge that drives knowledge growth. On the one hand, the students' cultural background as technology users affects their cognitive judgment of productivity. On the other hand, as the students gained more (cognitive) knowledge, their judgment of productivity continuously changed, and accordingly their approach to the assignment was extended from merely working out the problems as users at the interface level to analyzing the code in order to solve problems, as programmers.

CS education should be aware of the possibilities for culture clash and address it by designing courses to be fertile zones of cultural encounters through which students would be motivated and enabled to cross their cultural boundaries toward the academic CS culture. Course designers should strive for understanding the students' web of significance, and utilize the interest of the students in technology products to motivate their engagement in CS practice.