

# **A Visualization Environment for Computer Architecture**

Thesis for the degree DOCTOR OF PHILOSOPHY

Submitted to the Scientific Council of the

Weizmann Institute of Science

Rehovot, Israel

By Cecile Yehezkel

Advisors

Prof. Mordechai Ben-Ari & Prof. Tommy Dreyfus

May 2004

## **Abstract**

This thesis describes research whose goal is to evaluate a visualization environment intended to assist in teaching the subject "Computer Architecture and Assembly language." A second goal is to formulate guidelines for the development of visualization environments in this domain.

Assembly language is the closest programming language to the machine. Every instruction in this language is the description of an operation performed by the hardware during its execution. A dynamic display of each step of the execution of an assembly language program shows how the computer hardware components work together. This is a complex mechanism which is hard for students to understand. These difficulties have motivated the development of a visualization environment EasyCPU intended to describe the dynamic processes and information flows involved in program execution. The environment presents a schematic model of the components of the computer: the CPU (central processing unit), the memory and the input and output units. The environment provides an animation of the information flows among the different units, together with tools needed to develop basic assembly language programming skills (an editor, a simulated compiler, a linker and a debugger). A set of activities appropriate to the model presented to the student in the visualization was developed to support the teaching process.

The development of a visualization environment requires that the developers engage in analysis and decision-making concerning the conceptualization of the model that will be presented to students. It may be a model of an actual computer, a simplified model or an hypothetical model. Each model may be presented at different levels of abstraction (transistors, logic gate and so on...). Therefore the level of abstraction and the degree of complexity of the

model have to be taken in consideration during the conceptualization of the model, and have to be appropriate for the target population and learning materials. Additional decisions in the visualization environment development process are related to the selection of methods to facilitate interaction between the student and the environment. These analysis and design activities form an integral part of the development of the environment.

Research was conducted to examine and evaluate the implications on the learning process of the solutions adopted. The lack of an appropriate framework to define the characteristics of visualization environment motivated the construction of a new taxonomy for program visualizations, emphasizing the didactic and cognitive aspects of the visualization environment. This taxonomy was then used as a research tool to analyze the findings subsequently obtained during on the evaluation phase of the research.

The main part of the research concerned evaluation of the visualization environment within an educational framework. It was divided into three independent phases. The first two phases were based upon two of the main categories defined in the taxonomy: the presentation methods and the activity styles. These phases were aimed at examining the students use of the visualization component of the environment to performed an analysis activity (error detection in a program) and a development activity (write a new program). The third phase was based on the categories concerning content modeling. It focused on the evolution of students' mental models of the computer architecture as they were influenced by interactive work with the visualization environment.

The research was carried out in an educational framework over a long period. To cope with the analysis of the massive amount of data, a methodological approach was implemented based on visual analysis. This methodological approach is likely to be of use in other research projects that evaluate visualization environments, simulation tools and graphic interfaces.

The first phase of the research focused on how students used the environment, and examined whether it influences the methods students use to perform an analysis activity. Two groups of students who studied the unit "Computer Organization and Assembly Language" were involved: an experimental group worked with the EasyCPU environment and a control group worked with the professional Turbo Assembler (TA) environment. The findings showed that TA and EasyCPU students used different methods to test the program. The EasyCPU visualization renders the displayed information and the program's execution comprehensible, and the students' better control of the tools provided by the environment and the feedback they gained from the visual display encouraged them to refine the testing procedure and to proceed in a convergent manner toward the correct solution.

This process contributed to the attainment and the refinement of a higher cognitive level of testing skills, and strengthened the comprehension of program execution.

The second phase of the research focused on a development activity. The use of the visualization components and their contributions to the task execution were examined. The students were asked to write a program within the visualization environment to implement the control system of a simple elevator. The students had to define for themselves the behavioral model of the elevator that describes the components of the system, as well as the different states of the system that result from the interaction between the user and the elevator. The goal of the activity was to implement a conditional model in the visualization environment that performed as specified by the behavioral model. The findings showed that the visualization environment helped the students translate the behavioral model into the conditional model and enabled them to test the conditional model they implemented and its consistency with the required behavioral model.

The third phase of the research focused on the model presented to the student in the visualization and its influence on student's mental models. The visualization features of the EasyCPU environment enable students to develop a viable model of the computer architecture by visualizing dynamic interaction between the units during scenarios of instruction execution. Learning in the visualization environment provides students with skills enabling them to predict the execution process of an instruction. These skills are required to trace a program's execution in which a sequence of many instructions is involved: know how to identify the relevant information, know how to examine the processes and be able to understand them. The visualization makes the model concrete and therefore students to relate to it as a "unique and absolute" model of the computer. In some cases they misinterpreted the representation and attributed to the model features that are not implemented in the visualization. Their tendency was to combine components from models in different levels of abstraction which they learned about earlier.

These findings point out the influence on the learning process of conceptual models presented to the student by a visualization environment. The conceptual models presented in a visualization environment have to be designed through an evaluation procedure to ensure that the model is correctly interpreted by the students. The research approach described here can be used by developers engaged in similar conceptual design processes.