Inside the Computer: Visualization and Mental Models

Cecile Yehezkel, Mordechai Ben-Ari, Tommy Dreyfus
Department of Science Teaching, Weizmann Institute of Science, Israel

ntcecile@wisemail.weizmann.ac.il

1 Introduction

The EasyCPU environment was developed for teaching computer architecture to high school students of computer science. As a result of experience during the development, it was decided to perform research on the *mental models* that arise in the students as they learn. These mental models come from attempts by the students to make sense of the *conceptual models* presented to them by the software tool and their learning materials (textbook and exercises). These conceptual models concern the various components of the computer and their interconnections. We wished to investigate the relationship between the conceptual models that were presented and the mental models of the students. We conjectured that experience of the students as end-users of computers would be the basis of an initial non-viable mental model that the conceptual model would have to overcome.

2 Theoretical background

The first researcher to use the concept of mental models was Craik (1943), but the development of the theory of mental models was advanced by the later birth of cognitive science. The renewal of interest in mental models dates from the publication of Johnson-Laird (1983) and Gentner and Stevens (1983). Johnson-Laird (1983) saw mental models as a way of describing the thought processes of people as they solved deductive problems. Gentner and Stevens (1983) collected the work of several researchers on this topic, and claimed that mental models supply people with a means of understanding the functioning of physical systems. Norman's paper in this collection (Norman, 1983) defines a mental model M(T) as a conceptualization of a target system T by a *user*. According to him, this conceptualization defines the manner in which the user will carry out an operation on T. The mental model M(T) develops over time as a result of the interaction of the user with the target system T. M(T) is distinct from C(T), the conceptual model, which is the conceptualization of T by the *developer*.

Norman (1983) claimed that mental models do not have to be accurate, but they must be functional. The factors which affect the development of a mental model include: the technical background of the user, previous experience with similar systems and the structure of the user's own thought patterns. The user has a tendency to make up general rules that seem to fit all target systems. The mental model serves to guide decisions: the user can "run" the model to predict the outcome of an operation on the system. In effect, the model enables the user to mentally exercise the system.

When studying computer architecture, a student may not interact directly with a target system T, but rather with a *learning model* of the system L(T) that is presented by the visualization system on a computer platform. Therefore, it is possible that the resultant mental model M(T) may be different that it would have been had the student worked directly with T.

3 The EasyCPU environment

The EasyCPU environment is based on a simplified model of an 8-bit version of the Intel 80X86 microprocessor family (Yehezkel et al., 2001). EasyCPU has two modes of operation: a basic mode and an advanced mode. The basic-mode enables the novice student to learn the syntax and semantics of individual assembly language instructions (Figure 1). The visualization shows the main units of the computer: CPU, memory segments, and elementary I/O. The units are connected by control, address, and data busses, which are animated during the execution of an instruction. The student can simulate a single instruction's execution and follow the data transfer between the CPU's registers and the data flow between the units.

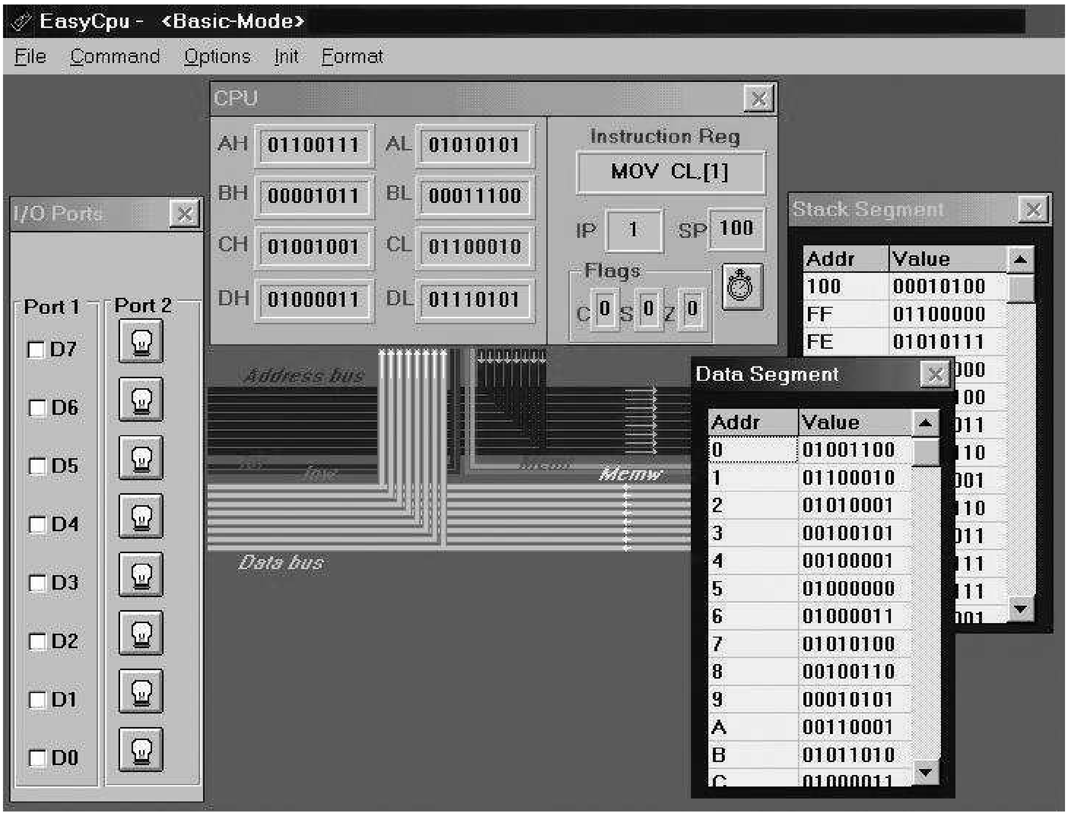

Figure 1: A screenshot the EasyCPU Basic Mode during the execution of `mov cl, [1]`

4 How can we describe the computer model?

A computer can be described both from a *static* viewpoint—the different units and a map of their interconnections, and from a *dynamic* viewpoint which describes the interaction and data transfer between the units. The conceptual model is based on four units (CPU, memory, input and output), and is composed of:

- The layout of the units and their interconnections.
- The method of transferring data along the connections.
- Standard interaction scenarios based on bus cycles.

The first two form the static viewpoint and the third describes the dynamic viewpoint. From the specific instruction being executed and its operands, you can predict what operations will be performed and describe them in a scenario.

5 Research

The research was carried out on a class of eleven tenth-grade students. They studied a one-year 90-hour course in computer architecture and assembly language programming. The students were evaluated on the basis of a project that they developed.

The mental models of the students were investigated at two points of time: first, after studying the material theoretically from the textbook but before they were exposed to the EasyCPU environment, and second, after carrying out tasks using the environment. The research tools were two tasks, a pretest and a posttest, in which the students were asked to describe both the static viewpoint (the topology of the interconnections between the units) and the dynamic viewpoint (six scenarios describing data transfer as a result of executing specific instructions). They were asked to describe the system model both graphically and in written text alongside the graphs.

The results were summarized in a table, with a row for each student, showing the six dynamic models that the student drew. This enabled us to identify the characteristic topologies of the drawings and to assign to each one a category defining the mental model associated with the topology. This gave us a mental model for each student at the pretest and then again at the posttest. In order to validate the results, the written text was also analyzed, searching for textual evidence that would support or contradict the categorization of the graph that was drawn.

We found that the students develop mental models that can be categorized into the four different topologies shown in Figure 2. For this target system, the CC model is the correct

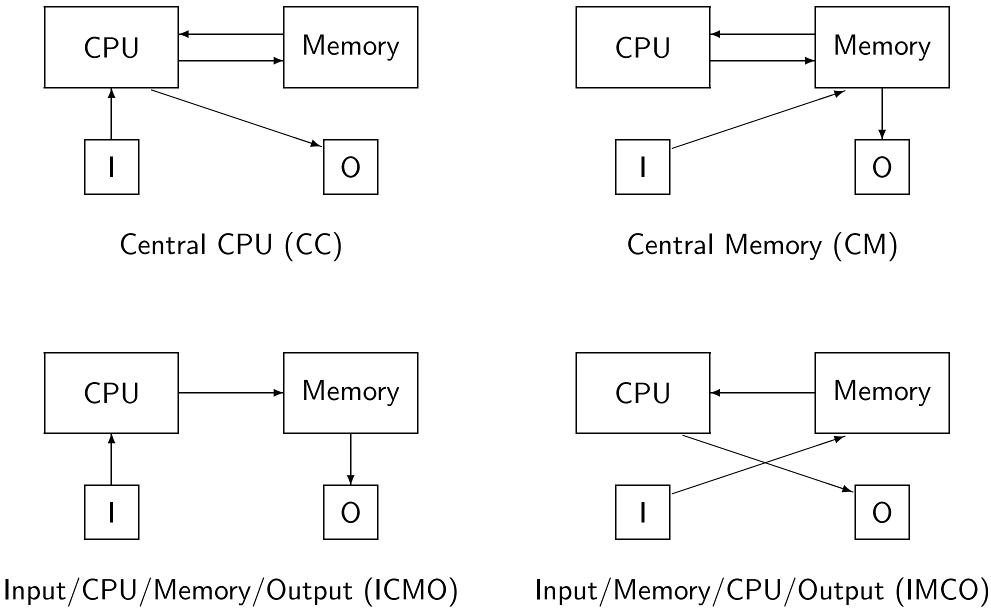

Figure 2: The four topologies

one, since all memory accesses and input-output are performed via the CPU. The CM model incorrectly assumes that I/O is done directly to memory. We will use the term ICO model for the two variants, one with the memory inserted before the CPU (IMCO) and one with the memory inserted after the CPU (ICMO). These models stresses input and output operations, downplaying the role of the main memory.

On the pretest, five of the eleven students chose to draw systems consistent with the ICO model, two more drew CM models, and only four drew the correct CC model. We interpret this as occurring because the students are used to interacting with the computer as end-users, which would tend to encourage the construction of a mental model M(T) consistent with the ICO model. The end-user interacts with the computer through I/O and experiences the computer as a processor of information, streaming from input to the CPU for processing and then to the output. As one student said:

> The data and instructions are entered through the input to the CPU, during this process the data is translated to the binary base and the instructions are translated

> to machine code. Then the CPU executes the operation. The result is transferred to the memory and then displayed onto the output.

Students who described CM models were influenced by what they learned about the stored-program architecture: A typical explanation by these students is:

> The instructions are entered through the input to the memory. ... The memory executes the instruction and controls the information flow.

On the posttest, after the students had interacted with the L(T) model presented by the visualization, all the students drew models consistent with the correct CC model. This supports the view that the visualization was critical in enabling the construction of a viable mental model, a process that did not occur from textbook learning alone.

6 Conclusions

After studying theoretical materials, the majority of the students still held mental models that had been influenced by their experience as end-users. The visualization feature of the basic mode of the EasyCPU environment enabled the students to develop a viable model of the computer architecture by visualizing dynamic interaction between the units during scenarios of instruction execution. This paper describes part of a larger research project that points 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.

References

Kenneth Craik. *The Nature of Explanation.* Cambridge University Press, Cambridge, 1943.

Dedre Gentner and Albert L. Stevens, editors. *Mental Models.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

Philip N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness.* Harvard University Press, Cambridge, MA, 1983.

Donald A. Norman. Some observations on mental models. In Dedre Gentner and Albert L. Stevens, editors, *Mental Models*, pages 7–14. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

Cecile Yehezkel, William Yurcik, Murray Pearson, and Dean Armstrong. Three simulator tools for teaching computer architecture: EasyCPU, Little Man Computer, and RTLSim. *Journal on Educational Resources in Computing*, 1(4):60–80, 2001.