Developing Analytical and Synthetic Thinking in Technology Education

Ilja Levin, E. Lieberman

Abstract
One of the most prominent characteristics of modern society is the increasing number of students acquiring technology education. An important question that must be dealt with, regarding this phenomenon relates to the nature of an appropriate technology education. A thorough examination of prevalent trends indicates that cultivating analytical skills constitutes an essential feature of science education, while within the framework of technology education mainly synthetic skills are being cultivated. Analytical thinking deserves little attention in processes of teaching technology, and is not adequately stressed in processes of constructing design skills. Apparently it seems that the different curricula adopted in science and technology education emanates from the inherent differences between research methodologies in science as opposed to design in technology. Whereas analytical thinking is typically related to the scientific process, synthetic thinking manifested in planning, building and developing is an essential part of design processes. However, several stages requiring analytical thinking can be identified in the design process. These stages mainly characterize the initial process and include analyzing the task, the selection of an appropriate model, formalization, etc.

Technology is viewed, within the conceptual framework of our research, as a discipline based on two types of thinking: synthetic and analytical, occurring both in the realm of practice (in the real world) and the realm of theory (using symbolic representations of the real world). The hypothesis examined in this research relates to the desired interactions between the two types of thinking, as well as to the manner of their integration in processes of teaching and learning. We hypothesize that integrating the above mentioned types of thinking might enhance the efficiency of technology instruction. In order to examine the hypothesis an interactive learning environment - SMILE (State machine Interactive Learning Environment) has been developed. SMILE enables the designing of computer systems by using different representations (Flowcharts, State Machine Graphs, etc). SMILE enables both the designing of computer systems (synthetic thinking) and the analyzing of systems by using formal procedures (analytical thinking). The main question that has been posed in this research focuses on examining the nature of the desired interaction between analytical and synthetic thinking (both in real design processes and in their symbolic representations) in planning teaching and learning activities by using the SMILE system.

Keywords
Technology Thinking, Learning Environment, Computer Systems Design.

A Model of Knowledge Building and Knowledge Using in Computer Systems Design

A map classifying disciplines into four categories using two axes: Analytic/Synthetic and Symbolic/Real had been presented by L. Owen (1998). Disciplines that are characterized as Analytical and Symbolic are concerned with finding and discovery, whereas disciplines characterized as Synthetic and Real are concerned with making and inventing. This map allows Owen to classify disciplines into the realm of practice and the realm of theory.

The above-mentioned mapping had been adopted to serve as a basis for the development of a new model, the purpose of which is to plan an integrative learning environment in the realm of “Computer Systems Design”. This learning environment uniquely integrates the Analytic/Synthetic and Symbolic/Real axes.

While the Symbolic/Synthetic area of the diagram relates to abstract notations of the system designed, the Real/Synthetic area involves skills required in the actual design process. And whereas the Analytic/Real area is concerned with analyzing systems both before and after the design process, the Analytic/Symbolic area deals with generalizations, theoretical aspects and the auto-formalization of the knowledge required in the design process. The integrated learning environment developed in the framework of this research enables the construction of all four types of skills.

Fig. 1 illustrates the above-described rationale of the learning environment.

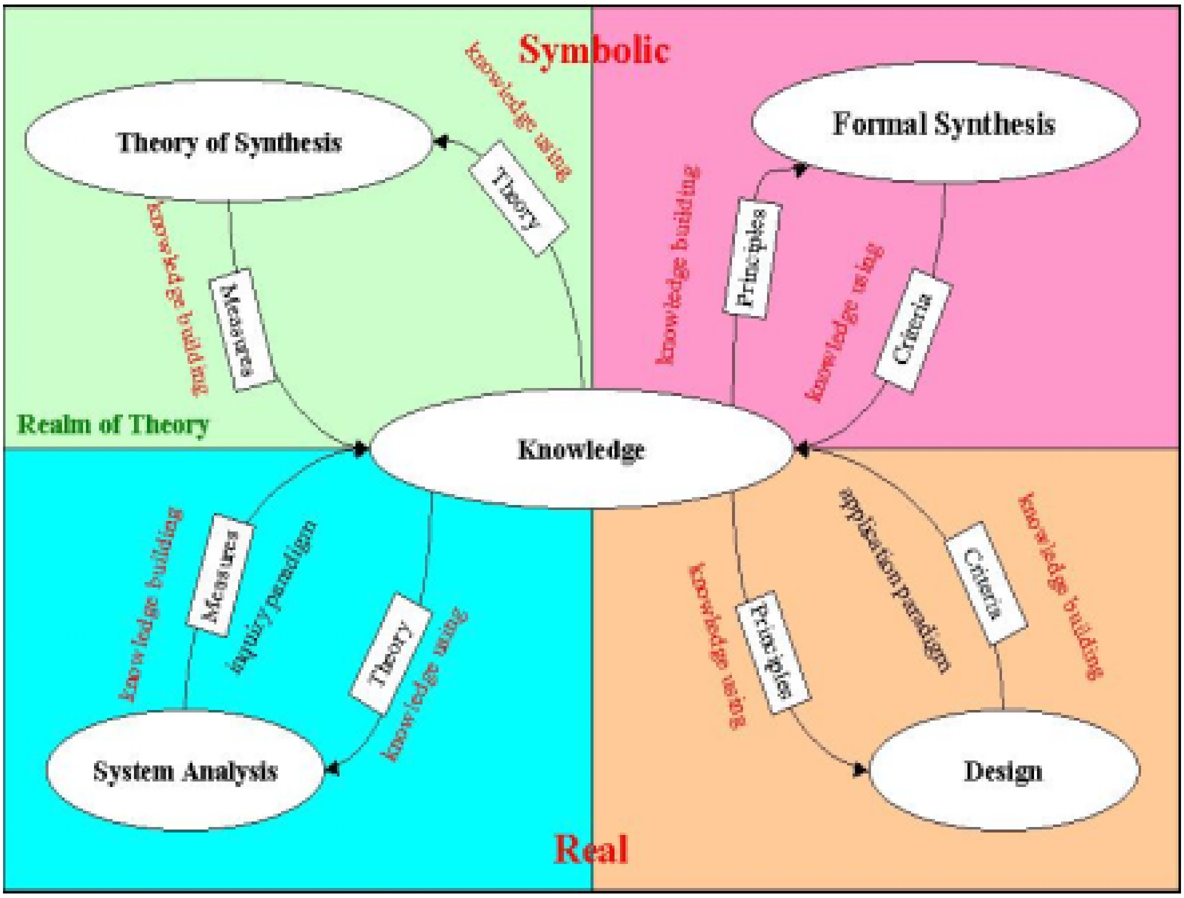

Figure 1. Model of Knowledge Building and Knowledge Using in Computer Systems Design.

Two modes of thinking can be distinguished in computer design learning: *synthetic* and *analytic*. The *synthetic* thinking involves the ability to plan, create and design computer systems, whereas *analytical* thinking involves the ability to formalize the design process in order to optimize it, according to various criteria. These two modes of thinking require different strategies and learning activities in the process of their construction. For example, learning activities usually utilized in constructing design skills are based on Computer Aided Design (CAD) systems, hardware design languages (VHDL, Verilog, Altera, Xillinx etc.) or on special design learning environments. *Analytical* thinking is cultivated by strategies requiring problem solving of formalization and optimization.

Both analytical and synthetic thinking can be manifested in either real design processes or in symbolic representations of these processes.

Formal (symbolic) models of Algorithmic State Machine (ASM) and Finite State Machine (FSM) are traditionally used in design processes serving for planning and implementing the actual design (synthetic thinking). Nevertheless these models also contribute to the optimization of the final design according to different criteria (analytical thinking). While the product of optimization processes is always a symbolic representation, the product of planning and implementing can be either an actual design in the real environment, or a symbolic representation underlining the design.

The above-mentioned paradigms were suggested as conveying very different cognitive approaches to computer systems design: the programming and the design paradigms (Levin and Mioduser, 1996). Both paradigmes have been used in the learning environment developed in the framework of our research: ASM representing the programming paradigm, and FSM representing the design paradigm.

Learning activities cultivating analytical and synthetic thinking skills can be based on formal (symbolic) models of ASM and FSM. Examples of possible symbolic learning activities are:

- Combining several ASMs (FSMs) into one ASM (FSM) (Baranov, 1994).
- Decomposing of one ASM (FSM) into a network of several components (Baranov, 1994).
- Investigation of the influence of certain changes made to ASM, on the FSM-representation of the system (Mioduser and Levin, 1996).
- Implementation of the same computer system using the ASM and the FSM descriptions, and analysis of the received results (Mioduser and Levin, 1996).

Interactive Learning Environment

The interactive learning environment SMILE (State Machine Interactive Learning Environment) was developed in the School of Education at Tel-Aviv University. SMILE-environment was developed for implementation of the above-mentioned ideas, i.e.:

SMILE enables the development of both analytical and synthetic thinking skills either in real design processes or in symbolic representations of these processes.

SMILE is capable of controlling real equipment.

SMILE is capable of providing a wide set of the above-mentioned symbolic learning activities.

SMILE supports both the programming and the design paradigms of teaching computer based control system concepts (Levin and Mioduser, 1996).

SMILE comprises two Editors: the Algorithmic State Machine Editor (ASM Editor) and the Finite State Machine Editor (ASM Editor), having each its display window. The SMILE-environment screen shot is shown on Fig. 2.

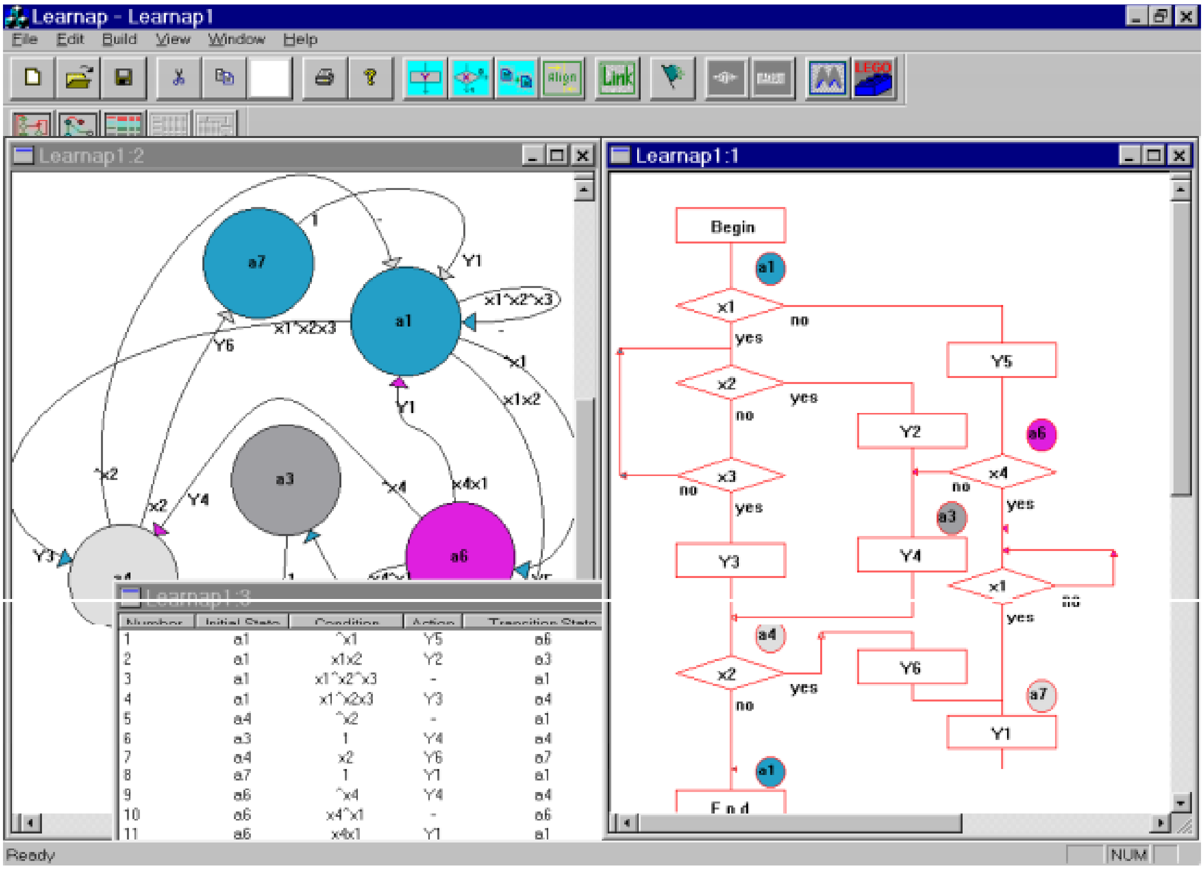

Figure 2. The SMILE-environment screen shot.

The environment enables the students to design a system both in the form of FSM by the FSM Editor, and in the form of ASM by the ASM editor. When the system is designed in one of these editors it can be visualized simultaneously in the windows of both editors. Any changes made in one of the editors cause relevant changes to appear not only in its window but also in the other editor window.

Both of the State Machine Editors ensure a number of very powerful features for describing virtually any type of a State Machine. For example, a student can represent a State Machine hierarchically, i.e. a State Machine having a large number of states can be described in multiple levels. For debugging a State Machine, two very important features are available in the Editors. The first is an animated simulation where any step in the control process is depicted by a change in color of the ASM (and/or FSM) symbolic states, which can be seen in the appropriate display windows. The second feature of the Editors is a possibility to study the control process watching the behavior of real equipment.

It should be noted that, when the system description is debugged, the resulting State Machine file could be obtained in the accepted hardware description language forms (VHDL, Verilog). Such a file can be further used for a real design.

Conclusion

We have proposed an approach, enabling the construction of both analytical and synthetic thinking skills, essential in all processes of designing computer systems. This approach allows us to integrate the required thinking skills in various learning activities, according to the types of technological problems to be solved. Using these skills in processes of either designing the actual system, or formulating and transforming symbolic representations of the design optimizes the learning process. State Machine Interactive Learning Environment (SMILE) has been developed to serve these processes.

References

Baranov. S. (1994). *Logic Synthesis for Control Automata*. Kluwer Academic Publisher, Dordrecht/Boston/London.

Levin I., Mioduser D. (1996). A Multiple-Constructs Framework for Teaching Control Concepts, *IEEE Transactions of Education*, 39 (4), 488-496.

Mioduser D., Levin I. (1996) Cognitive-conceptual Model for Integration Robotics and Control into the Curriculum, *Computer Science Education* Special Issue: Robotics in Computer Science & Engineering Education, 7(2), 199-210.

Owen Ch. L. (1998, January). Design research: building the knowledge base. *Design Stadies*, 19 (1), 9-20.