

Do Projects in Information Systems Design Really Facilitate Meaningful Learning in Computer Science in High School?

Nayif Awad

Abstract—This paper deals with creating a learner-centered environment in the context of teaching information systems design (ISD) in high school. The participants were 40 students who prepared individual graduating projects, for example, an information system for a small business or a medical clinic. Data were collected through observations in the class, interviews with students, analysis of their products, and portfolios. The study revealed that providing the students with the opportunity to choose projects that were meaningful to them personally or that related to their family business fostered their motivation in completing the task. The students greatly improved their programming skills related to ISD; however, many of them only partially completed all of the design phases, for example, inquiring into customers' needs and planning. The current situation in schools shows that students learn and practice using procedural knowledge, whereas achieving conceptual knowledge is a long-term process.

Index Terms—Design, information system, project.

I. CONCEPTUAL FRAMEWORK

A. Contextual Learning

The term contextual learning deals with learning that relates to a learner's diverse life contexts such as at home, during leisure time, social or environmental activities, or in the workplace [1]. Contextual learning is reality-based learning that provides students with opportunities to make meaning of their disciplinary knowledge and solve problems within a real-world context [2], [3].

B. Project-Based Learning (PBL)

Project-based learning is a strategy that places the learner in a meaningful learning situation that is focused on the solution of a problem taken from a real situation. The learner takes the initiative to construct knowledge and effectively develop a solution to the problem by providing the necessary resources, guidance and opportunities for exploration [4]. Ref. [5] point out that "Computer science projects are often complex, marrying design, human communication, human-computer interaction, and technology to satisfy objectives ranging from consolidation of technical skills through provoking insight into organizational practice, teamwork and professional issues, to inculcating academic discipline and presentation skills."

C. System Design Models

Students at school often develop their projects in information systems according to a linear model that consists of the following stages: 1) identifying a problem or need; 2) researching and setting specifications for the required solution; 3) generating alternative solutions and choosing the optimal one; 4) planning; 5) implementing; 6) testing and evaluating; and 7) improving. More professional models in the literature, such as the Three-Phase Dual-Loop (TPDL) model [6], [7], which is illustrated in Fig. 1, present an iterative process that often involves moving backwards and forwards from one point to another, and highlights the use of conceptual versus procedural knowledge in the different design phases.

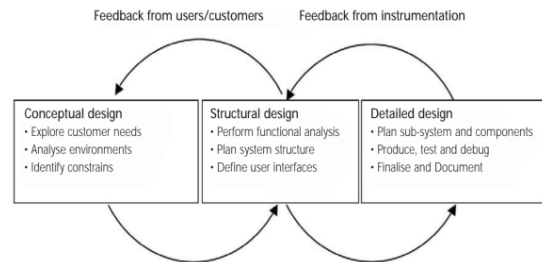


Fig. 1. The three-phase dual-loop (TPDL) system design model.

We will show how the TPDL model could help in analyzing the advantages and limitations of teaching design in school, with a focus on the example of information systems design.

II. THE RESEARCH

A. Research Objectives and Guiding Questions

The research was guided by the following questions:

- 1) How do students deal with designing an information system?
- 2) To what extent can students develop their projects according to the TPDL model?

B. Participants and Setting

The participants were 40 12th-grade students (aged 17-18) specializing in computer science in two high schools in northern Israel. The students prepared an individual graduating project in information systems design and completed a final oral exam.

C. Methods of Data Collection

The research adopted a qualitative approach in order to expose as many learning patterns as possible. Data collection

Manuscript received May 5, 2015; revised July 17, 2015.
 Nayif Awad is with Ben Gurion University of the Negev, Israel (e-mail: awad_nayif@yahoo.com).

included class observations—documenting students’ activities in the class, interviews with a sample of 12 students, and analyzing students’ final projects. The data analysis was performed through three concurrent flows of activity: data reduction, data display and thematic interpretation [8]. It involved the selection of fragments relevant to the specific issues above. These fragments were transcribed and analyzed by the author. Categories and their attributes emerged from a detailed sententious analysis of the data.

III. FINDINGS

A. How the Students Selected Their Project Topics?

Eight of the 40 students selected a topic they had already dealt with in the small assignments they had prepared in 10th or 11th grade, seven accepted the topic the teacher had suggested to them, and four said that they decided to work on a particular topic because it sounded interesting to them without knowing too much initially about the subject matter. For example, a student decided to develop an information system for a travel agency because she thought it would be interesting to deal with flights, hotels and international tourist attractions. The remaining 21 students decided to develop an information system on a topic that was of personal interest to them or to their parents’ vocation or business. For example, a student worked on an information system for a small garage where he was working part-time as an apprentice.

The fact that the students worked on topics that interested them personally very positively influenced their motivation to cope with the task. In the interviews, the students had comments such as:

“I did not join my parents on vacation... instead I stayed home to work on my project.”

“I worked all night on my project.”

B. How the Students Addressed the Conceptual Design of Their Information System?

The students were required to carry out an investigation of a customer’s needs and prepare a document called a ‘project charter’ consisting of: 1) background of the project; 2) customer’s needs; 3) changes or improvements required in the existing system; 4) aims and objectives in terms of detailed measurable results; 5) criteria of success; 6) consequences of failure; 7) constraints and factors that might limit the planning and how to accommodate them; and 8) possible risks, their probabilities and how to overcome them. Some of the students were very serious about investigating a customer’s needs. For example, a student who developed an information system for a travel agency visited several travel agents and interviewed local employees. The student learned about the types of information handled by the travel agent, such as flights, hotels and car rentals. He also studied the interface screens the users require in working with the system, as well as the kinds of reports and printed documentation the system must provide. In contrast, other students only partially investigated the topic they were working on. Of the 40 students, 10 prepared a project charter that sufficiently covered all the required points, 16 prepared a document that only partly answered the

demands and needed improvement, and 14 students prepared a very superficial or incomplete document.

Since this stage of the project had only little to do with ‘hands-on’ work in using computers, the students who majored in computer science regarded this stage of their project as a formal matter rather than an essential part of the system development. In addition, many felt it was legitimate to estimate or visualize the data for their system, as they used to do in 10th and 11th grades when solving small programming exercises. This point also appeared in the way the students sketched their systems’ designs, as described in the following section.

C. How the Students Handled the Information System’s Structural Design?

The systematic design of a software system is important for obtaining reliable and efficient software, and for enabling communication between all individuals or teams involved in the system development, implementation or future upgrades. The design of an information system takes place by drawing several diagrams and charts that present, for example, the architectural structure of the software system, its class inheritance, data flow, state machines, sequence diagrams, database tables and relationships. Yet, did the students in the current study internalize this view of a project in computer science?

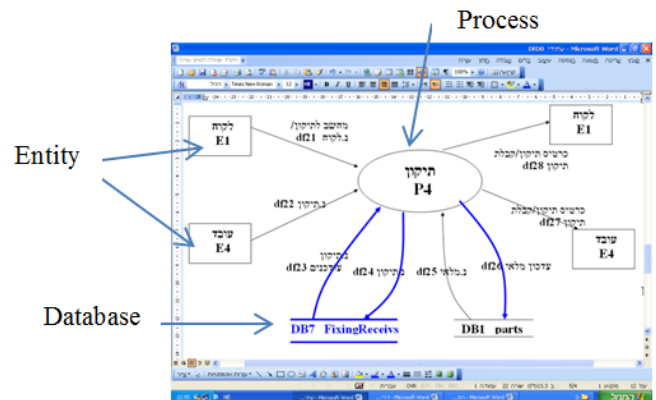


Fig. 2. A data flow diagram (DFD) sketched by a student as part of designing the information system.

The students sketched different charts and diagrams using a word processor (Fig. 2). Moreover, they captured the entire screen using the PrtScr key and pasted it as a picture into their portfolios. In reality, today’s designers use different sophisticated design tools such as Computer-Aided Software Engineering (CASE) and OOAD (Object-Oriented Analysis and Design) tools. This is an indication of the phenomenon that schools perceive whereby the essence of projects was teaching programming, while the aims of fostering students’ design skills and using advanced technological tools for design received less attention.

D. How the Students Performed the Detailed Design Phase Mainly Dealing with Programming?

The major part of the students’ work on their projects was programming the information system that was comprised of two stages: creating the databases for the system using Access (Fig. 3); and programming all the logical functions, computations, user interface screens and reports using Visual Basic (Fig. 4).

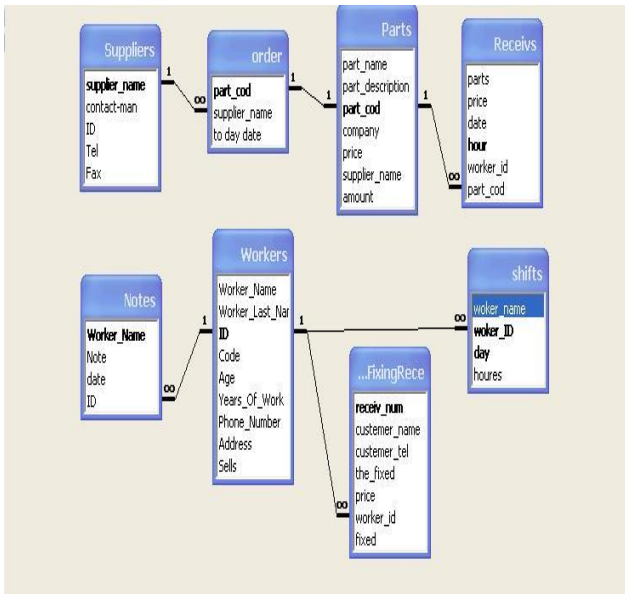


Fig. 3. Typical database structure created by access.

Fig. 4. Typical form created by visual basic.

In the initial programming stages of the project, the students progressed very slowly and some felt frustrated (they had a limited programming experience in 10th and 11th grades). The majority of the students ignored many of the good programming rules or guidelines, such as handling using variables, functions or procedures, and verifying the validity of data entered. Students made the following comments in the interviews:

“The computer went completely crazy.”

“The teacher didn’t ask me to do that.”

“Programming today is totally different from previous years.”

The students’ programming skills improved gradually as expressed by their use of modular programming methods, handling Visual Basic error messages, and re-writing more efficient or elegant programs. Below are examples of statements of students made in the interviews:

“I really understand the error now.”

“I made the same error again because I was in a hurry.”

“I write down every error message... I take notes... the next time it appears, I will solve it faster...”

E. Students’ Final Scores

The students’ final scores on their projects (on a scale of

0-100) are marked based on an oral exam by an external examiner from the Ministry of Education. Of the 20 students in one school, eight were granted a score ranging from 90-100 (excellent), three from 80-89 (very good), five from 70-79 (good), and four from 60-69 (poor). The distribution of scores in these examinations in the other school was quite similar, and both schools present a common picture of students’ achievements in their final projects on information systems design in Israeli high schools. These findings indicate that the graduating project on information systems design is a demanding task and not all students excel in it.

IV. DISCUSSION AND CONCLUSIONS

Information systems design is likely to provide a good platform for creating a constructivist technology-based learning environment in school. However, the design of a real-life system is often a complicated task for high school students because it requires the integration of conceptual knowledge, mainly during the phase of defining a system’s objectives and general structure, and procedural knowledge, for example, in the detailed design, implementation and testing phase.

The common situation in schools is that students learn and practice mainly procedural knowledge, whereas accumulating conceptual knowledge is a long-term process. Indeed, the current research highlighted that the conceptual design phase might appear to be one of the most challenging parts for students inexperienced in carrying out a design project because this task has to do mainly with exploring peoples’ needs, desires and expectations. As students proceed to detailed design phases, including implementing and programming, the task becomes more concrete to them since it is essentially about procedures and instrumentation.

To encourage learning and knowledge construction, it is important to merge the teaching of design concepts with the teaching of programming by engaging students in open-ended design tasks at the very early stages of learning computer science. It is recommended that educators give students small, open-ended design tasks and gradually increase the scope and complexity of these assignments. After all, the student’s designing skills are being developed after cycles of experimentation, interaction with instrumentation, collaboration with peers and reflection.

REFERENCES

- [1] J. Dewey, “How we think: A restatement of the relation of reflective thinking to the educative process,” *Heath*, 1963.
- [2] D. Karweit, *Contextual Learning: A Review and Synthesis*, Baltimore, MD: Center for the Social Organization of Schools, Johns Hopkins University, 1993.
- [3] J. R. Savery, “Overview of problem-based learning: Definitions and distinctions,” *The Interdisciplinary Journal of Problem-Based Learning*, vol. 1, no. 1, pp. 9-20, 2006.
- [4] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar, “Motivating project-based learning: Sustaining the doing, supporting the learning,” *Educational Psychologist*, vol. 26, pp. 369-398, 1991.
- [5] S. Fincher and M. Petre, “Project-based learning practices in computer science education,” *Proceedings – Frontiers in Education Conference*, vol. 3, pp. 1185-1191, 1998.
- [6] B. S. Blanchard and W.J. Fabrycky, *Systems Engineering and Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1998.

- [7] K. Miettinen, "Design: Structure, process, and function: A systems methodology perspective," in *Philosophy and Design, From Engineering to Architecture*, P. E. Vermaas, P. A. Kroes, A. Light, and S. Moore, Ed. Springer Netherlands, pp. 217-231, 2008.
- [8] M. B. Miles and A. M. Huberman, *Qualitative data analysis: an expanded sourcebook*, London: Sage, 1994.

University with acknowledgement of excellence in his final paper. Currently he is a lecturer of mathematics and computer science at the Academic College for qualifying teachers in Sakhnin and a coordinator of ICT specialization.

Mr. Awad has an experience in teaching software engineering at the high school level for more than 10 years.



Nayif Awad is a PhD student in science and technology education at Ben Gurion University of the Negev. His research deals with promoting an integrated learning of STEM subjects at rich computer-based learning environments. He received his B.Sc. degree in computer engineering from the Technion, Israel Institute of Technology, and his M.A. degree in science and technology from Ben Gurion