

## **Problem Solving with an Icon Oriented Programming Tool: A Case Study in Technology Education**

Jari M. Lavonen, Veijo P. Meisalo and Matti Lattu

### **Introduction**

In 1995, the authors started the Empirica Control for Technology Education (*ECTE*) project in the Department of Teacher Education at the University of Helsinki with the purpose of widening the possibilities for creative problem solving in technology education. First, an icon-oriented visual programming tool was developed to teach computer control technology, along with an appropriate computer interface. Authors developed the new programming tool to minimize the need for direct guidance from the teacher and to reduce the need for rote memorization of programming rules. It was also designed to allow for constructive and creative activities by the students. Along with the software and hardware, handbooks for creative problem solving in control technology were written and in-service training was organized to support teachers in their efforts to develop creative problem solving through technology education programs in Finnish comprehensive schools.

The development of this new computer control system led to an interest among the authors in finding out how pupils learn the basics of control technology and programming and how they creatively solve problems within the context of computer control.

### **Creative Problem Solving**

The terms “problem” and “problem solving process” have been defined in many ways (e.g., McCade, 1990; Fisher, 1990, p. 100; Higgins, 1994, pp. 20-21). For example, the terms “designing,” “trouble shooting,” “solving textbook problems,” and “experimenting” are sometimes used interchangeably with “problem solving”. In this study, the focus was on creative problems, meaning ill-defined and multifaceted real world problems that pupils seek and find in their environment (cf., Higgins, 1994, pp. 35-57; Lewis, Petrina & Hill, 1998). Therefore, it was important to analyze the whole environment; one must be aware that a problem must exist before it can be solved. When problem solving is creative, ideas or products produced during the problem solving process are original and appropriate (Fisher, 1990, pp. 29-31). Effective problem solving is a process that consists of various stages. These may include formulating the problem, recognizing facts related to the problem, setting goals, ideating or generating alternatives, evaluating ideas, choosing the most promising solution, and the testing and evaluating of the problem, recognizing facts related to the problem, setting goals, ideating or generating alternatives, evaluating ideas, choosing the most promising solution, and the testing and evaluating of the

Jari M. Lavonen (Jari.lavonen@helsinki.fi), Veijo P. Meisalo (Veijo.meisalo@helsinki.fi), and Matti P. Lattu (Matti.lattu@helsinki.fi) are staff members in the Department of Teacher Education, University of Helsinki, Finland.

solution (see Table 1 below; Fisher, 1990, p. 39; De Luca, 1993; Higgins, 1994, p. 19). The problem solving process is not linear and does not strictly follow any particular rules. Structured approaches often miss the whole point of creative problem solving.

Because of how the human mind works to create new ideas, pupils need to apply thinking that is critical, systematic, analytic, and vertical as well as thinking that is creative, intuitive, divergent, and lateral in their problem solving. The emphasis in modern education has often been exclusively on critical thinking. Of course, even critical thinking is needed in problem solving, especially in the recognition of facts related to the problem and in the evaluation of the ideas. The need for various special approaches to promote creative thinking arises from the limitations of the behavior of the mind as a self-maximizing memory system (de Bono, 1970). Therefore, various idea generation techniques or ideation models are valuable (Smith, 1998). Consequently, the outcomes of creative problem solving activities depend on the creative processes and ideation techniques that are learned and applied. Furthermore, there are attitudinal (interest, motivation, and confidence), cognitive (knowledge, memory, and thinking skills), and experiential (familiarity with content, context, and strategies) factors that influence problem solving processes (Fisher, 1990, p. 112). For example, non-judgmental, positive feedback and the acceptance of all ideas, even absurd or impractical ones, are important in all creative group processes for generating non-trivial alternatives (Higgins, 1994, p. 119). In Table 1, some key features that are typical of creative group processes are presented (cf. Runco & Okuda, 1988; Fisher, 1990, pp. 97-129; Higgins, 1994).

Various ways of emphasizing (creative) problem solving in a learning environment have been suggested (Grabinger, 1996, p. 665; Dooley, 1997; Hill, 1999). A common feature of these approaches is to place pupils in the midst of a realistic, ill-defined, complex, and meaningful problem, with no obvious or "correct" solution. Pupils act as professionals in small groups and confront problems as they occur, with no absolute boundaries, insufficient information, and a need to settle on the best possible solution by a given date. In other words, learning is authentic (e.g., Lafer & Markert, 1994) in that it involves real-world problem solving situations and is self-directed and reflective. This kind of problem-centered approach empowers the pupils to take responsibility for their learning by allowing them to define what they need to learn and to identify the resources needed. The teacher's role is that of a facilitator in the learning process.

Sellwood (1991, pp. 4-6), De Luca (1993) and Williams and Williams (1997) argued that creative problem-solving activities are an integral part of design and technology education, in contrast to instruction that is a step-by-step process, engaging students in reproducing artifacts in an environment dominated by the teacher. Some researchers suggest even more forcefully that creative problem solving is the core content and an important teaching method of design and technology education (Lee 1996; Wu, Custer & Dyrenfurth, 1996). Therefore, in the ECTE project special attention was given to learning materials

**Table 1.**  
*Key features typical of creative problem solving in groups.*

Features of creative problem solving	Indications of the features
1. Suitable atmosphere for creative problem solving	Group members: <ul style="list-style-type: none"> <li>• Trust one another and believe in the power of group work.</li> <li>• Are motivated, active participants and apply their creativity to the problem.</li> <li>• Have a positive and constructively critical attitude about the ideas presented by other group members.</li> </ul>
2. Knowledge of the problem solving process and its application in fostering creativity.	Group members: <ul style="list-style-type: none"> <li>• Identify and focus on essential aspects of the problem.</li> <li>• Recognize and find the facts related to the problem.</li> <li>• Set goals and have a vision for solving the problem.</li> <li>• Know and apply ideation models and are able to generate new, original ideas.</li> <li>• Appreciate the ideas of others and can provide positive feedback, resulting in further development of the ideas.</li> <li>• Think intuitively, creatively, and divergently, as well as think systematically, critically, and analytically.</li> <li>• Know and use techniques for evaluating ideas</li> <li>• Separate ideation from evaluation.</li> <li>• Put ideas into practice through modeling, evaluation, and further development.</li> </ul>

that would promote creative problem solving in a group and to various ideation techniques applicable to control technology projects.

### The Programming Tool

Various investigations have been conducted in learning environments where computers, interfaces, and construction kits or building blocks are used in control technology for hands-on projects that require problem solving in groups (Parkinson, 1999). Several studies have been conducted in the Lego/Logo (Lego TC Logo) learning environment. Their aim has typically been to find out about the learning of various skills, qualities of social interaction, problem-solving approaches, and the attitudes of pupils towards their study (Lafer & Markert, 1994). It has been emphasized (e.g., Järvinen, 1998) that the syntax sensitivity of the Logo language makes it cognitively complex, resulting in programming tasks that are difficult and frustrating for pupils. Moreover, only Legos can be connected to the Lego interface and the selection of sensors is rather limited. Although much work has been done already, more development work and research are needed to ascertain the effects of microcomputer-assisted approaches to teaching control technology in various learning environments.

In the ECTE project, the authors developed the *Empirica Control for Windows 95*, an icon-oriented programming tool. The hypothesis of the developers was that a visual programming language based on icons makes programming easier than with text-based languages. When programming with languages like *Visual Basic* and *Logo*, one has to be very careful with the program structure and the spelling of the code words. With these types of languages, the skill of using the tool, not problem solving, often becomes the main focus. With *Empirica Control*, the user simply drags icons to a program diagram instead of typing programming code. This visual approach to the language makes the programming process much more concrete. The user simply chooses command icons by pointing to them with a mouse and clicking; hence the icons are linked with lines to form a structure like a flow chart.

Several developers have used the “mini-language” approach (e.g., Brusilovsky, Kouchnirenko, Miller & Tomek, 1994) whereby a pupil learns to program using the mental analogy of the control of an “actor” in the programming process. This sort of approach was applied in the *Empirica Control* system. While running a program, a blue ball moves along the flow chart, indicating which of the commands the computer is currently processing. The pupil can imagine that writing the program is the same as writing rules for that blue ball. The blue ball is analogous to an actor in the context mentioned above. All the parameters for commands are set from dialogue windows, which means that few details have to be memorized. The *Empirica Control* gets data from the environment via the *Empirica I/O Interface* connected to the RS-232 serial port of a computer. The interface has been designed especially for educational institutions. For example, its digital outputs are able to deliver currents up to one ampere, which make it suitable for the direct control of DC motors. In addition, over 20 different sensors can be connected as analog inputs. Thus, the system has the versatility to allow for a wide variety of solutions to a given problem.

A user’s guide (Lavonen & Meisalo, 1997a) and a handbook called *Technology* (Lavonen & Meisalo, 1997b) were written to help teachers organize learning environments in which pupils can learn the basic commands, principles, and skills needed to operate the *Empirica Control*. The first section of the handbook is reference material that includes information about the basics of programming with the *Empirica Control*. The second section explains the basics of technological systems, such as the concepts of open versus closed loops and the elements of input, process, and output (see Hacker & Barden, 1988, pp. 47-56). The second section also includes examples, which express the essential role of control technology in home, industry, and society. The third section deals with broader projects with special attention given to creative problem solving in general as well as practical problem-solving models. Furthermore, some idea generation techniques are introduced. The general aim of this section is to help the pupil to discover how the learning environment can help in planning, designing, constructing, programming, testing, redesigning, and evaluating.

It is obvious that more research and development efforts are needed to better understand how to introduce learning environments with a problem-

solving approach that are effective in technology education programs (Lee, 1996). Creative problem solving has been the leading principle in the design of learning environments and it was a significant influence on the research goals of this study. The main purpose was to evaluate the nature of the pupil's creative problem solving processes in a learning environment equipped with learning materials developed in the ECTE project. The principal research questions were:

1. Is it possible to organize problem-centered and creative learning in a learning environment using the software and hardware developed in the ECTE project?
2. What are the indications of problem-centered learning in the learning environment?
3. What are the indications of pupils' creative roles and behavior in problem solving within control technology projects?

### **The Empirical Study**

This study can best be described as a qualitative case study since the researchers selected for closer examination a typical example from a small number of other examples. The case study approach was chosen because it gives the best possibilities for closely following the problem-solving process in a particular learning environment, and consequently to raise questions for further research and development. Case study research "seeks to understand specific issues and problems of practice" (Merriam, 1988, p. 23) through a detailed examination of a specific group of people, a particular organization, or a selected activity. Naturally, such an approach does not allow any broad generalizations to be made. However, this restriction was accepted at this explorative stage of the research and development process.

#### *The teaching experiment*

The experiment was organized at a teacher training school located in a metropolitan area in Finland during the spring term of 1998. A total of 34 eighth-grade (14-year-old) pupils in three separate groups attended an elective technology course arranged by a science teacher in the school. The technology theme was new to all the pupils. The pupils worked in three study groups, in randomly assigned pairs, for 20 hours. A computer equipped with the Empirica Control, the *Empirica Control Guidebook*, a *Lasy* robotics kit, a set of cables, and a set of lamps were available to each pair.

A male teacher with considerable teaching experience in science was trained to use the Empirica Control software and the Empirica Interface during a three-day in-service training workshop. During the workshop, the teacher also became familiar with the technology theme to be used in the experiment. After the workshop the teacher studied the *Empirica Control Guidebook* (Lavonen & Meisalo, 1997a) and a technology education handbook (Lavonen & Meisalo, 1997b). He also practiced with the software and hardware. The basic principles of creative problem solving were familiar to the teacher beforehand. Before the teaching experiment began, the teacher discussed and planned the experiment with one of the researchers.

The course began with a two-hour introduction during which the main operations of the Empirica Control system were presented. In the second period, lasting 10-12 hours, the pupils solved technological problems using the system. The teacher and other pupils, as well as the user manual, provided support. During this period, practical tasks were selected to familiarize pupils with various programming structures such as if-then and loop structures, as well as typical technological processes such as automatic switching. For example, the project introducing automatic switches was formulated in relation to an everyday situation, by telling a story of how tenants of an apartment building constantly forgot to turn the lights off in the basement. The pupils had to develop various solutions to this authentic, open-ended problem. Other projects included designing a rotating advertising booth, an automatic gate, an elevator, and a robot. During the last six to eight hours of the teaching experiment, the pupils also had the possibility to create problems of their own choosing.

#### *Collecting the data*

To ensure the validity and credibility of the research, various approaches to applying triangulation in the data acquisition process were used (Cohen & Manion, 1986, pp. 254-271). This involved video recordings of the pupils' problem-solving processes, observer's field notes, teacher interviews, and documentation of the pupils' computer program files.

The field notes were written in the classroom and completed immediately after the field research according to the principles of non-participant observations (Cohen & Manion 1986, pp. 120-147). Observation as a research method has been criticized because of its subjectivity and because it allows researchers to observe only the external behavior and actions of participants. Therefore, the researchers observed and videotaped the pupils in three separate groups so that inconsistent findings could be determined. The videotapes allowed the activities of the pupils to be observed several times and were a principal means of collecting data. Though videotaping can affect the students' activities, the teacher felt that it was not a factor in this experiment. Available resources limited the recording to one hour in each of the three groups. One of the researchers, in consultation with the teacher, selected representative pairs of students for video recording from each of the three groups. The field notes confirmed that the activities and success of the selected pairs did not differ from those of the majority. To get the most relevant data for this study, the recordings were made during the second period, a time when the students worked on small open-ended problems suggested by the teacher. All computer program files created by the pupils were collected.

One of the problems that the students were given was to "Create a program and the wiring, which turns the fan on when the temperature is over 27°C and the button is pushed. The system should turn the fan off when the temperature drops below 27°C or the button is released." Two example solutions to this problem are shown in Figure 1.

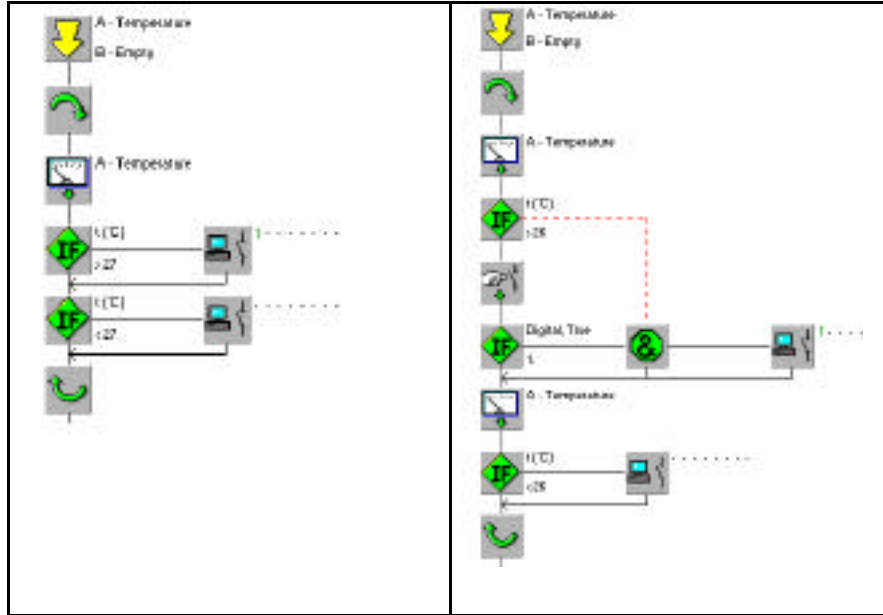


Figure 1. Sample student solutions to programming problem.

The teacher was interviewed using an unstructured interview (Cohen & Manion, 1986, pp. 291-314). Notes were written during the interview and finalized immediately after its completion. The interview provided information on the goals of the course and the teacher's behavior during the experiment. Furthermore, the observations that were unclear or confusing were validated through the interview. It was also possible to compare the videotaped examples to the remainder of the course. A qualitative interview outline was prepared beforehand to support the interview and reduce interview subjectivity. The outline consisted of the following five questions:

1. What goals did you have in mind when you planned the technology projects?
2. Ask about problem-centered learning and creativity if the teacher does not otherwise discuss them.
3. What do you think about reaching the goals?
4. Ask about achieving the goals considering the knowledge and various skills (creative, programming, ...) of the students.
5. Can you please analyze a) your own and b) pupils behavior during the project?
6. What do you think about the learning environment used in this project?
7. Ask about the physical (software and hardware) and pedagogical nature of the learning environment and how a teacher changes the environment during the course or between the courses, if the teacher does not otherwise discuss them.



The transcribed field notes and the teacher's interview covered nine standard pages.

#### *Analyzing the data*

Preliminary data analysis was started immediately after the initial data were collected. However, the comprehensive analysis was performed after all data collection was completed. The intensive data analysis began by first reviewing the purpose of the study: to find positive and/or negative indications of pupils' problem-centered learning and indications of pupils creative roles and behavior in problem solving within control technology projects. After that, the researchers read the field notes, reviewed the teacher interview record, and observed the videotapes twice, while discussing preliminary findings with each other. One researcher recorded the main verbal and nonverbal events and the other researchers validated the notes on the basis of the video recordings. These data enabled the researchers to observe patterns, propose explanations, develop categorical definitions, and to create more differentiated and integrated text (Huberman & Miles, 1994, p. 433).

The data analysis was structured into categories and subcategories according to the main objectives of the study. The first category, problem-centered learning and its environment, was divided into four subcategories:

- The nature of problem-centered activities
- The nature of pupils' activity in a learning environment
- The nature of teachers' activity in a learning environment
- The nature of how students learn computer programming.

The second broad category was organized according to the key features typical of creative problem solving, as presented in Table 1. Further information from the creative process was obtained by analyzing program files created by the pupils.

### **Results**

During the teaching experiment, the pupils successfully solved their control technology problems with the new programming tool and hardware developed by the ECTE project. What follows are the positive and negative indications of pupils' problem-centered learning and indications of creative processes are described based on the video recordings' notes, the field notes, and the teacher interview. Representative responses are presented below.

#### *Problem-centered approach*

As evidenced from the notes taken on the videos, it was apparent how the pupils solve technological problems with the programming tool without formal teaching: "Although the pair (A) faced a problem, they still continue to work on the task". The field notes revealed how pupils acted as professionals in the small groups and agreed on the assignment of tasks: "The pairs assigned activities, e.g., one works with the interface and Lasy kit and the other programs". The pairs worked in various ways. In some groups one student mainly worked on the program and the other constructed the models. In other groups, students changed their roles as they worked on the problem. Apparent in the videos and notes was



how intensively the pupils were engaged in problem solving: "When a pupil from pair C came to see how pair B was succeeding, the latter continued their work without any interruption." As a confirmation, the teacher spontaneously remarked in the interview: "The work was problem centered all the time." One central deficiency in the problem-centered approach was that the teacher set the starting point of the projects. On the other hand, the notes stated: "In addition to the given problems, pair (B) varied and extended its solutions".

It was clearly seen in the videos that the teacher had mainly taken the role of a questioner, attempting to clarify the pupils' ideas. The teacher asked questions like: "What are you doing? What is your aim? What have you done up to now? How is your model and program working? What are the inputs and outputs in your program"? Though this approach worked with some students, it did not with others. In some cases, the teacher seemed to give direct advice on how to edit the program, taking away the opportunity for the students to solve the problem on their own.

The notes and videos showed that during the problem-centered projects, it was reasonably easy for the students to become familiar with the programming tool. The teacher confirmed this: "The students learn quite easily to setup the program, start programming, and make the connections to the interface. The visual programming tool helps in programming." On the other hand, if the pupils had been taught the basics of programming ahead of time and if the handbook had been introduced, their autonomy in solving the problems would likely have increased. The lack of planning and programming skills appeared in both the observers' notes and the teacher's comments: "Low-achievers especially have problems in long-term independent effort requiring planning and programming." The notes from the video also revealed that the pupils did not find the facts they needed in the handbook: "A pupil (in pair A) glanced through the handbook but could not apply it to his problem." Moreover, the pupils in pair A hesitated to ask for help when faced with difficulties. It might be possible that they did not recognize the need for assistance as long as they were able to proceed by trial and error. Their need for help was not obvious from observing them.

In conclusion, the data indicate that pupils' work was based on a problem-centered approach. The teacher confirmed this finding: "The most central aim in the learning environment was to develop students' problem-solving skills by allowing the students to face technological problems with no ready-made recipe."

#### *Creative processes*

There were many notes that indicated that the atmosphere was open, promoting creativity: "Atmosphere in the classroom is high and emancipated; the group easily reaches an agreement on the assignment of tasks; the pupils discuss and think aloud when programming, discussion is democratic; the pupils discuss the logical operations, listen to each others' opinions (not underestimating others' ideas)." The notes indicated that pupils trusted their team members and the power of teamwork; they thought positively and were motivated.

Almost no indications of the pupils' systematic planning (recognizing and finding facts related to the problem and setting goals and visions) or systematic activities to generate alternatives before the active construction phase were observed, even though planning tools and ideation techniques were introduced in the handbook. They seemed too eager to start the constructive work. It was seen in the notes and in the videos how the pupils began their work immediately after receiving the task assignments: "The pupils (pair B) began their work immediately after studying the problem. One pupil explored the programming environment while the other connected devices (a push-button and a motor) to the interface. Then they began their work by creating the program. A discussion then took place." However, an unplanned project quite often leads to ineffective work: "The work of pair (A) is of a trial-and-error type." On the other hand, the planning and pondering of various alternatives was observed at a later stage in the problem-solving process and the solutions of the groups were unique. All the programs had an analog temperature input and a loop structure. One third of the programs had an if-structure and half of them included a logical and. One group decided to utilize the trigger feature. It was concluded that the visual programming tool promoted individual, unique solutions. Two examples of team solutions were shown in Figure 1.

The debugging and evaluation of a program appeared to be easily accomplished by students with the blue ball icon that represented the movement of the program through the flow chart while it was running. The students could easily observe which of the commands was currently being processed. Moreover, it is easy to see how certain commands had an effect on devices such as a gate or a thermostat. The videos and notes revealed how the pupils continued to debug their program, modify parameters, add commands, and so forth: "A pupil (in pair B) starts to program, puts his finger on the screen, follows the blue ball, and looks at the fan. The fan does not rotate. The pupils discuss ...". On the other hand, the pupils could not independently decide when the project was complete. The notes indicate that: "The pupils waited for the teacher's acceptance of their project." The researchers interpreted this to mean that pupils lacked the necessary tools and skills for evaluation.

### **Discussion**

The Empirica Control system was developed to minimize the need for rote memorization of programming rules and to allow room for creative problem-solving activities. These goals were the focus of this study. From the data collected in this study it was clear that it is possible to organize control technology learning activities so that pupils can solve problems and simultaneously learn to autonomously construct computer programs for controlling processes. If the activities are carefully selected, they introduce pupils to the basic processes in control technology (cf. De Luca, 1992). In the teacher's opinion, one possible reason for the user friendliness of the Empirica Control program was the visual and schematic nature of programming. It was clear that creating programs in a graphic flow chart and allowing the program flow to be easily observed as it was executed are important.

Systematically teaching the programming skills to the students before they were presented with the problem may have been a more effective approach. Combining the learning of the programming approach with solving the actual problem resulted in a very complex process (Taylor, 1991). At times the pupils had difficulty proceeding with the solution of the problem, in spite of the teacher's indirect guidance. It seemed that if the students had become well acquainted with the Empirica handbook ahead of time, it would have ensured greater autonomy in solving the problem. This appeared to be especially the case with learners who had difficulty with independent learning. The videos clearly show pupils trying to glance through the handbook to find help in solving their problems. If they had studied the handbook first, it seems that they would have understood more about the structure of programs and consequently they would have been able to apply this knowledge to solving the problem with which they were confronted. Interestingly, it is possible that the high level of user-friendliness that the Empirica system provided might have actually promoted a trial and error approach instead of one that was more systematic. Since it seems to be difficult to integrate handbooks and other written materials in a problem-centered learning environment, the teacher might begin by combining open-ended problem solving and study of the handbook with short, easy problems. This could effectively introduce the concepts and the vocabulary while keeping a context of realistic practice. On the other hand, the teacher argued that directly teaching from the manual is contrary to the notion of student-directed learning.

The teacher succeeded very well in assuming the role of a tutor, giving pupils open-ended problems and asking questions to clarify the pupils' ideas. The pupils could also ask for help or ask the teacher questions. Nine times out of ten the teacher did not give direct answers but responded with additional "how" and "why" questions, or suggested the direction of thought the pupils might follow. It is challenging for the teacher to create an environment in which students can have sufficient time to learn new concepts well enough to apply them to the tasks at hand. The teacher described his work with: "The problem centered approach was very difficult for me, because I had to think hard all the time about how to avoid leading and how to ask indirect questions."

According to the field notes and videos, the pupils worked quite autonomously, but mainly "by trial and error." There was little evidence of reflective thinking. Systematic planning of the project or execution of plans was not observed. First-hand observations indicated that the lack of planning generated most of the difficulties experienced by pupils. All the difficulties pertained to programming. It is quite obvious that the pupils did not know programming well enough to apply it to the solution of the problem. The video showed how the teacher tried to introduce a stepwise analysis of the programs by analyzing the pupils' programs when tutoring, but pupils did not seem able to adopt this analytical method. Either the pupils did not understand the significance of such thinking or there was a breakdown in their communication with the teacher. Therefore, it was hypothesized that it is important to teach, either directly or otherwise, more about programming, at least in the beginning. For example, effective planning tools or strategies (e.g., utilizing the flow

diagram or making a list of inputs, processes, and outputs) for the program design could be very useful in the design of complex programs (cf. Gustafson & Rowell, 1998, p. 160). The Empirica Control software is itself a planning tool, because it creates block diagram illustrations of programs and their progress when executed. On the other hand, linear strategies do not easily help pupils to generate several possible solutions to their problems (Welch, 1999). It would be interesting to investigate what effect the pupils' systematic planning has on their problem solving, although the teacher thought that such planning would be boring to the students.

The lack of guidance and the nature of the task assigned may explain the lack of ideation activities. The course did not include any instruction related to creative processes or methods. Therefore, pupils were most likely unaware of the basic principles of how to engage in creative work. In addition, the problems in the task assignments were typically very specific to technology rather than general in nature. In some cases, the problem was defined using technical words like "wiring" and "button" rather than words that may have been more easily understood. The task was approached in a technical way instead of as a functional problem. The pupils may not have seen the problems in the context of the technological world around them, and consequently could not start to solve them from their base of experience (cf. Runco & Okuda, 1988, pp. 211- 220). This was also the opinion of the teacher: "In the beginning of the technology course, special attention must be given to the technological world around us. The students have to be familiar with the examples of the feedback systems and loops etc. in real life situations." The problem solving experience lacked authenticity for the students. It was concluded that problems should be presented in authentic contexts and adequate time should be allocated at the beginning for the processing of ideas and the planning of solutions.

Overall, it appears that the pupils' work in this study was problem centered but it was not very creative. Indeed, creative processes were almost completely missing. In the approach used, students were not given the opportunity to formulate problems themselves even though such experiences are one of the most important phases in problem solving (Sapp, 1997, pp. 282-298). The approach did not encourage students to think of many possible solutions to the problems and then select the proper solution (cf. Amabile, 1996, pp. 88-89). Instead, they seemed to proceed with the first solution that surfaced and then apply it in a trial and error manner.

It seems that various ideation techniques (e.g., brainstorming and analogous thinking) need to be taught to students if they are to be successful in developing creative solutions to problem-centered projects (Smith, 1998, pp. 107-133). They must learn the facts connected to the background of the problem. This means that pupils have to collect relevant information about the problem and the processes that might be applied to its solution. Programming skills, even if they are working within a graphical environment like that used in this study, are needed if pupils are to realize their ideas. Pupils should also be familiar with creative ways of thinking such as how to think positively, give constructive criticism, ask relevant questions, and assist other pupils in developing their

ideas. Various evaluation techniques must also be learned for each stage of the project.

### References

- Amabile, T.M. (1996). *Creativity in context*. Boulder, CO: Westview Press, 88-89.
- Brusilovsky, P., Kouchnirenko, A., Miller, P., & Tomek, I. (1994, June). Teaching programming to novices: A review of approaches and tools. In *Educational multimedia and hypermedia, 1994. Proceedings of ED-MEDIA 94—World Conference on Educational Multimedia and Hypermedia* Vancouver, British Columbia, Canada. ERIC Document Reproduction Service No. ED388228
- Cohen, L., & Manion, L. (1986). *Research methods in education*. London: Croom Helm.
- de Bono, E. (1970). *Lateral thinking; Creativity step by step*. New York, N.Y.: Harper and Row.
- De Luca, V.W. (1993). Survey of technology education problem-solving activities. *The Technology Teacher*, 51(5), 26-30.
- Dooley, C. (1997). Problem-centered learning experiences: Exploring past, present and future perspectives. *Roeper Review*, 19(4), 192-196.
- Fisher, R. (1990). *Teaching children to think*. Oxford: Basil Blackwell Ltd.
- Glaser, B.G., & Strauss, A. L. (1967). *The discovery of grounded theory*. Chicago, IL: Aldine Publishing Company.
- Grabinger, R.S. (1996). Rich environments for active learning. In R. Jonassen (Ed.) *Handbook of research for educational communications and technology*. London: Prentice Hall International.
- Gustafson, B.J., & Rowell, P.M. (1998). Elementary children's technological problem solving: Selecting an initial course of action. *Research in Science and technological education*, 16(2), 151-164.
- Hacker, M., & Barden, R. (1988). *Living with technology*. New York: Delmar.
- Higgins, J.M. (1994). *Creative Problem Solving Techniques: The Handbook of New Ideas for Business*. Florida: New Management Publishing Company, Inc.
- Hill, J.R. (1999). Teaching technology: Implementing a problem-centered, activity-based approach. *Journal of Research on Computing in Education*, 31(3), 261-280.
- Huberman, A.M. & Miles, M.B. (1994). Data management and analysis methods. In N.K. Denzin & Y.S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 428-440). Thousand Oaks: SAGE Publications.
- Järvinen, E.M. (1998). The Lego/Logo learning environment in technology education: An experiment in a Finnish context. *Journal of Technology Education*, 9(2), 47-59.
- Lafer, S., & Markert, A. (1994). Authentic learning situations and the potential of Lego TC Logo. *Computers in the Schools*, 11(1), 79-94.

- Lavonen, J., & Meisalo, V. (1997a). *Empirica Control käyttöohje [Operation manual for Empirica Control]*. LUONTI Project. University of Helsinki. Department of Teacher Education.
- Lavonen, J., & Meisalo, V. (1997b). *Teknologia [Technology]*. LUONTI Project. University of Helsinki: Department of Teacher Education.
- Lee, L.-S.S. (1996, March), Problem-solving as intent and content of technology education. Paper presented at the 58th Annual Meeting of the International Technology Education Association, Phoenix, AZ. ERIC Document Reproduction Service No. ED391959.
- Lewis, T., Petrina, S. & Hill, A.M. (1998). Problem posing - Adding a creative increment to technological problem solving. *Journal of Industrial Teacher Education*, 36(1), 5-35.
- McCade, J. (1990). Problem Solving: Much more than just design. *Journal of Technology Education*, 2(1), 28-42.
- Merriam, S.B. (1988). *Case study research in education: A qualitative approach*. San Francisco: Jossey-Bass.
- Parkinson, E. (1999). Re-constructing the construction kit - Re-constructing childhood: A synthesis of the influences which have helped to give shape and form to kit-based construction activities in the primary school classroom. *International Journal of Technology and Design Education*, 9, 173-194.
- Runco, M., & Okuda, S. (1988). Problem discovery, divergent thinking and the creative process. *Journal of Youth and Adolescence*, 17, 211-220.
- Sapp, D.D. (1998). Problem parameters and problem finding in art education. *The Journal of Creative Behaviour*, 31(4), 282-298.
- Sellwood, P.A. (1991). The investigative learning process. *The Journal of Design & Technology Teaching*, 24(1), 4-12.
- Smith, G. (1998). Idea generation techniques: A formula of active ingredients. *The Journal of Creative Behaviour*, 32(2), 107-133.
- Taylor, K.A. (1991). *An annotated bibliography of current literature dealing with the effective teaching of computer programming in high schools*. Student research project. Indiana: Indiana University at South Bend. ERIC Document Reproduction Service No. ED 338217
- Welch, M. (1999). Analyzing the tacit strategies of novice designers. *Research in Science and Technological Education*, 17(1), 19-34.
- Williams, A., & Williams, J. (1997). Problem based learning: An appropriate methodology for technology education. *Research in Science and Technological Education*, 15(1), 91-103.
- Wu, T.F., Custer, R.L., & Dyrenfurth M.J. (1996). Technological and personal problem solving styles: Is there a difference? *Journal of Technology Education*, 7(2), 55-69