Effects of Using Model Robots in the Education of Programming

Attila PÁSZTOR, Róbert PAP-SZIGETI, Erika LAKATOS TÖRÖK
Kecskemét College, Faculty of GAMF, Department of Information Technology
Izsáki 10, 6000 Kecskemét, Hungary
e-mail: {pasztor.attila; pap-szigeti.robert; torok.erika}@gamf.kefo.hu

Received: August 2009

Abstract. In this article we try to show how new devices and methods can help in the education of programming. At Kecskemét College programmable mobile robots and instead of behavioral, the constructivist pedagogical methods were used. Our experiments have proved our hypothesis as the improved new methodical education using devices can give more practical programming knowledge, increases the attitude towards programming and helps to have positive programming self-image. The results of the experimental and control groups were compared at the beginning and at the end of semester, when the programming knowledge and motives were measured. During the learning process only the experimental groups used devices and new methods.

Keywords: education of programming, innovative devices, constructivist method, measuring of effects.

1. Introduction

Nowadays in Hungary lots of ways of technical education teaching IT as well as teaching programming has been paid great attention. It's very important for every student to develop his algorithmic thinking and problem solving abilities with computer. IT students can get better jobs easier when they have compatible knowledge in programming.

In Hungarian secondary public education the elements of programming and basic algorithms are not in compulsory curriculum. So there are lots of students who only during their high studies meet elements of programming. According to our experiences (Kiss and Pásztor, 2006; Pásztor, 2008) the students who have background knowledge from secondary studies can be more successful in their studies in programming at college. In our supposition the students who meet programming firstly at college can hardly have the ability of abstract thinking for problem solving.

In the first part of our article we introduce some approaches from theory of cognitive development of Piaget that solve as basic approaches. One of them is the neo-piagetion theory (see, e.g., Demetriou *et al.*, 1992). The neo-piagetion pedagogy could get successful results firstly with rearrangement of the curriculum and using of devices characterized of certain stages of development (Adey and Shayer, 1994; Csapó, 1999). The constructivist pedagogy built also on the Piaget's theories (Strommen and Lincoln, 1992), wants to create the possibility of individual knowledge-construction based on students' experiences. So our hypothesis is that we can help the student's period of formal operations and knowledge-construction in programming with using concrete, curriculum-compatible devices.

Following the introduction of structure and methods of subject adapted to our hypothesis we also introduce the results and interpretation of our study.

2. Theoretical Framework for Our Study

The wide applying of Information and Communication Technology in education gives a question: which ways of educational organization and form of work are needed for the productive teaching-learning process in a changed learning environment.

Similarly to the traditional educational environment, in education supported by information technology tools the teaching materials and educational methods are introduced to some kinds of teaching theory. The behavioral, cognitive and constructivist teaching theory discusses the learning process in different ways. The behavioral strategy underlines the facts, the cognitive aspect emphasizes operation of processes and the constructivism according to the real situations underlines the personal interpretations as original fundamental basic points. In the middle of behavioral and cognitive approach there are teacher and teaching material, the constructive one needs problem solving based on student activities in real life situations. According to the constructivist educational approach (Strommen and Lincoln, 1992) the student based on his own experience interprets the information effect on him. In this approach the student builds and constructs himself his own knowledge base. In this situation the students stands in the center of learning process and the teacher only helps, gives advises as a facilitator. The personal knowledge depends on physical and intellectual environment.

The changes in pedagogical concept of the abilities were strongly influenced by Piaget's model about the development of personality. Piaget described this development as a qualitatively instead of quantitatively distinguishable periods of life (Inhelder and Piaget, 1955). The first two stages of his theory, namely Sensory Motor Period and the Preoperational Period have a none-relevant role in our study because these stages are terminated at the age of 7. In the Period of Concrete Operations (7–11 years) the children's thinking becomes changeable. Children can recall a train in reverse order or can perform simple or multiple classification tasks. This period is a part of school years and there are many possibilities to operate with concrete objects or tools at the school. The Period of Formal Operations (11–16 years) extends the range of useable operations and contents. In this period children can appreciate and manipulate variables, formulate a hypothesis or comprehend possibilities and need of checkout (Turner, 1975). Although many of researchers level at theory of Piaget because of its rigorous age borders or idea of the universal nature of development, most of these researchers agree that the acquiring process of abstract operations can be supported by common actions with concrete tools or objects.

The neo-piagetian approach is based on Piaget's theory, but it refuses its rigorous age borders of it. This approach basically applies rearrangement the elements of curriculum to an order that corresponding to the students' age. In this approach the researchers often use tools that can ensure the necessary experience for the improvement of formal thinking. Applying this approach a permanent effect was measureable due to CASE (Cognitive Acceleration through Science Education) program leaded in the United Kingdom. The improvement of science thinking affected to the development of other cognitive fields (Adey and Shayer, 1994; Shayer, 1997). Similarly to this experiment the improvement of thinking abilities through enrichment of content and rearrangement of curriculum was turned out a success in Hungarian primary schools (Csapó, 1999).

Approaches orientated towards using objects and tools are presented in the teaching of programming. The educators and the researchers tried to concretize the functionality of abstract algorithms using screen objects from the early decades of the informatics' history. For this reason the turtle-graphic by Papert (Solomon and Papert, 1972), Karel the Robot (Pattis, 1981/1995) or the Spider World by Dalbey and Linn (1984) was developed. Busilovski *et al.* (1984) systematized the teaching approaches of early years and emphasized the important role of so-called mini-languages in the learning process of novice programmers. These languages were often developed for controlling tools towards teaching programming.

Applying such objects and devices that make possible to use concrete operations is often examined in educational researches. There are possibilities to observe and recognize the behavior or functionality of special robots at experience-level in Kindergarten (e.g., Levy and Mioduser, 2008) but most of these studies use objects and tools at the secondary schools or in the college education. Understanding the role, the functionality and the malfunctions of hardware made by students and software for controlling them were examined by Kamada *et al.* (2008) and by Kurebayashi *et al.* (2008). They primarily wished to strengthen their students' knowledge on machines controlled by computers. Their results showed that the secondary school students enjoyed to build and program the autonomous robots, furthermore the members of experimental group understood the activity of embedded systems significantly better than the control groups' members.

Some articles introduce how to use the devices of programmable LEGO and parts of Mindstorms robot family in education of programming. Sartatzemi *et al.* (2005) also examined in Greek secondary schools the possibilities of basic of programming with the help of robots. Although the low number of lessons their pilot-experiment successfully improved the student's programming knowledge and also introduced that understanding of concepts was often random. Wu *et al.* (2008) examined if programming education results change when instead of real robots they use screen-robots, which simulate real ones. There is no significant different between the two groups in understanding of programs but using of real robots the students could imagine the behavior of robots easier. The students working with real robots enjoyed studying much more.

The above-introduced researches prove that in learning of basic programming can help the using real devices and simple programming environment and education based on students' experiences. The themes can be easier activated and developed than applying traditional programming education.

3. Methods in Learning and Instruction

We introduced and applied a new subject named "Model Robot Programming" for IT students for the aim they can study the bases of programming in practice with using concrete, real devices. So they are able to acquire the essential programming structures visually with the help of LEGO NXT robots (Pap-Szigeti and Pásztor, 2008).

In Programming Model Robots subject we try to follow the principles, methods and working of constructive teaching approach. We use the traditional methods of teaching process only about twenty percents of the total lessons (teacher explanation, presentation). The new materials are introduced by examples, the needed functions and used programming languages are also introduced. During the following lessons the students solve programming problems and tasks in groups of two-three members cooperation each other. During these kinds of lessons the teachers instead of traditional knowledge transfer they support, motivate or coordinate students. We help the students with, e.g., useful collection of examples and sources in solving examples and problems and doing individual research works. The most important part of student work is to construct robot working by given program. In this project they work in small groups. Their project task is to write the algorithm and program and in the same time to build a special robot controlled by their program. They have to make the documentation, video and slides of their project and finally to present all of them to their classmates and answer their questions. The evaluation of their project is during the common made presentation. The project is evaluated by the teachers, other students and project group members.

In this subject the collaborative knowledge building realizes (Scardamalia and Bereiter, 2003) in which strategies of knowledge building increase the process of understanding/explanation and creation. The personal understanding is together with the social knowledge building (Stahl, 2006).

4. The Empirical Study

In this study we supposed that we simultaneously achieve activating and improving learning motives and acquiring the basic elements of programming by using real programmable tools.

H1: The learning process becomes more enjoyable when students use real programmable tools.

The feeling of knowledge-growing and the joyful learning can make the optimal situation of a very strong motive named "the flow" (Csíkszentmihályi, 1990). Furthermore the exercises at more and more difficult level can ensure the optimal challenging power, so they can activate the students' mastery motivation. This motivation plays a fundamental role in the process of skill development (Józsa, 2007).

H2: With using model robots we can strengthen our students' programming self-concept as a learning motive.

Experiences during learning process with robots and the successfully solved problems affect students' believes and position on themselves.

Table 1

Comparison of the two sub-samples' programming pre-test and previous programming self-concept

	Experimental average (st. dev.)	Control average (st. dev.)	t (p)
Programming test (%p)	44.6 (19.1)	41.3 (19.5)	1.36 (0.17)
Programming self-concept (%p)	47.2 (19.7)	46.3 (21.5)	1.57 (0.14)

H3: Exercises solved at the level of concrete operations affect students' programming knowledge and skills at the abstract operational level.

Learning to program a computer at the level of abstract operations is a very difficult for novice programmers. We suppose that mastering the abstract elements of programming is easier when students use a real tool and learn at the level of concrete operations.

4.1. *Sample and Measuring*

We conduct a study with control group to check our hypothesis. All of students who took part in this study had passed the course Programming I before. At the examined semester the members of experimental group ($n_{\text{exp}} = 73$) used LEGO NXT model robots during their learning process as we introduced in chapter 3. In the control groups ($n_{\text{ctrl}} = 76$) the learning and instruction was conventional.

For measuring our students' programming knowledge and skills we used advanced version of our antecedent test with 15 items (its Cronbach-alpha is 0.86). Most of items required a short answer, students had to explain, complete or debug some parts of simple programs. The same test was used in pre-test and post-test.

We used a questionnaire for measuring attitudes towards programming and programming self-concept. This questionnaire contained 17 items, mainly five-degree Likert-style sentences. Six sentences examined our students' programming self-concepts. These variables arranged into one factor (KMO = 0.87), that's way we made a collapsed variable without weight. This new variable was transformed into %point scale. In the pre-test we applied some additional questions for collecting background data. In the post-test the questionnaire contained additional Likert-style questions, which were focused towards enjoyment and difficulty of the learning during the semester.

The academic achievement was similar at that time when the two sub-samples passed the course Programming I. ($\chi^2 = 3.86;\ p = 0.38$). Their programming pre-knowledge and previous programming self-concept did not differ in the pre-test (Table 1). There was not a significant difference between our two sub-samples concerning the number of learning semesters they have learnt at secondary schools ($\chi^2 = 5.42;\ p = 0.27$). 46 percents of whole sample have not learnt programming at the secondary school.

We use Levene-test (F-test) in comparing standard deviations. The differences are non-significant in these comparisons.

Based on the similar pre-test data of two sub-samples we suppose that the post-test differences between the control and the experimental group are caused by educational effects. We present the measured changes in the next chapter.

4.2. *The Development of Two Sub-Samples*

The experimental semester was three and a half month long. Neither the experimental groups' ($x_{\text{pre}} = 44.6\%p$; $x_{\text{post}} = 47.9\%p$; $t = -1.23$; $p = 0.23$) nor the control groups' ($x_{\text{pre}} = 41.3\%p$; $x_{\text{post}} = 43.6\%p$; $t = -1.01$; $p = 0.34$) programming knowledge developed significantly during this semester. The correlations between the results of pre-test and post-test are similar in case of two sub-samples ($r_{\text{exp}} = 0.631$; $r_{\text{ctrl}} = 0.618$). There were small immediate effects of real tools and educational methods to the students' programming knowledge and skills.

However, the differences in changing of motives were important. Number of failed lessons was significantly less in the experimental group than in the control group ($\chi^2 = 3.22$; $p = 0.03$). Members of the experimental group enjoyed learning more than members of the control group ($x_{\text{exp}} = 3.47$; $x_{\text{ctrl}} = 2.96$; $t = 3.87$; $p < 0.01$). In accordance with this thing, the control group's students felt the material more difficult than the experimental group's students ($x_{\text{exp}} = 3.07$; $x_{\text{ctrl}} = 3.35$; $t = 1.96$; $p = 0.03$). The attitudes towards the teacher did not change notably ($x_{\text{pre-exp}} = 4.03$; $x_{\text{post-exp}} = 4.06$; $x_{\text{pre-ctrl}} = 4.03$; $x_{\text{post-ctrl}} = 4.12$; the differences are non-significant).

We found stable programming self-concept in case of the control group ($x_{\text{pre}} = 46.3\%p$; $x_{\text{post}} = 44.1\%p$; $t = -0.45$; $p = 0.66$), while the experimental group's programming self-concept developed significantly ($x_{\text{pre}} = 47.2\%p$; $x_{\text{post}} = 52.2\%p$; $t = -2.60$; $p = 0.01$). This difference in self-concepts is observable between the distributions of two sub-samples. This result suggests that despite of the shortness of experimental period the new tools and methods caused a notably development in students' programming self-concept. It is very important from the educational point of view, because the well-developed self-concept hardly effects the students' further school achievement (Helmke and van Aken, 1995; Niemivirta, 1997). By tracking of our students we wish to check whether the development of self-concept affects to further programming achievement or not.

5. Conclusions

Real tools and constructivist methods can help students to achieve the abstract element of programming at the level of concrete operations. Their knowledge can be based on their own experiences. This knowledge can be represented by a stronger, operationally enriched connection among the elements.

Students can experience a motivated, joyful learning situation while they use model robots. The direct motivational affects of educational toys usually are not permanent. However, in our study the new methods and tools, such as model robots, can contribute the development of learning motives, especially programming self-concept. It can strengthen and widen the students' motivational basis, and can affect to their further achievement in programming.

References

Adey, P és Shayer, M. (1994). *Really Raising Standards: Cognitive Intervention and Academic Achievement*. Routledge, London.

Brusilovsky, P. *et al.* (1994). Teaching programming to novices: a review of approaches and tools. In: *Educational Multimedia and Hypermedia. Proceedings of ED-MEDIA 94 – World Conference on Educational Multimedia and Hypermedia*, Vancouver, Canada, June 25–30. `http://www.eric.ed.gov/contentdelivery/servlet/ERICServlet?accno=ED388228`

Csapó, B. (1999). Improving thinking through the content of teaching. In: Hamers, J.H.M., Van Luit, J.E.H. és Csapó, B. (Eds.). *Teaching and Learning Thinking Skills*. Swets and Zeitlinger, Lisse, 37–62.

Csíkszentmihályi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper and Row, New York.

Dalbey, J., Linn, M. (1984). Spider world: a robot language for learning to program. Assessing the cognitive consequences of computer environments for learning (ACCCEL). Paper presented at the *Annual Meeting of the American Educational Research Association*, New Orleans, LA, April. `http://www.eric.ed.gov/contentdelivery/servlet/ERICServlet?accno=ED250178`

Demetriou, A., Shayer., M., Efklides, A. (1992, Eds.). *Neo-Piagetian Theories of Cognitive Development. Implications and Applications for Education*. Routledge, London and New York.

Helmke, A., van Aken, M.A. (1995). The causal ordering of academic achievement and self-concept of ability during elementary school: A longitudinal study. *Educational Psychology*, 87, 624–637.

Inhelder, B., Piaget, J. (1955). *The Growth of Logical Thinking from Childhood to Adolescence*. Routledge & Kegan Paul, London.

Józsa, K. (2007). The development of mastery motivation at age 10–16. Paper presented at *12th Biennial Conference for Research on Learning and Instruction*, Budapest, August 28 – September 1.

Kamada, T., Aoki, H., Kurebayashi, S., Yamamoto, Y. (2008). Development of an educational system to control robots for all students. In: Mittermeir, R.T., Syslo, M.M. (Eds.), *Informatics Education – Supporting Computional Thinking. Poceedings of the Third Conference on Informatics in Secondary Schools – Evolution and Perspectives*, ISSEP 2008, Torun, Poland, July 1–4, 63–74.

Kiss, R., Pásztor, A. (2006). *Programozható robotok felhasználása a programozás oktatásában* (Using programmable robots in teaching programming). Professional Day, Kecskemét College, Faculty of GAMF.

Kurebayashi, S., Aoki, H., Kamada, T., Kanemune, S., Kuno, Y. (2008). Proposal for teaching manufactoring and control programming using autonomous mobile robots with an arm. In: Mittermeir, R.T., Syslo, M.M. (Eds.), *Informatics Education – Supporting Computional Thinking. Poceedings of the Third Conference on Informatics in Secondary Schools – Evolution and Perspectives*, ISSEP 2008, Torun, Poland, July 1–4, 75–86.

Levy, S.T., Mioduser, D. (2008). Does it ”Want” or ”Was it programmed to...”? Kindergarten children’s explanations of an autonomous robot’s adaptive functioning. *International Journal of Technology and Design Education*, 18(4), 337–359.

Niemivirta, M.J. (1997). Academic achievement, self-concept and self-esteem: A longitudinal analysis of causal predominance. Paper presented at the *7th Eauropean Conference for Research on Learning and Instruction*, Athens, Greece.

Pap-Szigeti, R., Pásztor, A. (2008). Congruence examination of NXT robots in the education of programming at KF GAMF college. *Practice and Theory in Systems of Education*, 3(3–4), 33–40.

Pásztor, A. (2008). *How can IT be Taught Playfully with the Help of Programmable Robots?* INTED, Valencia.

Pattis, R.E. (1981/1995). *Karel the Robot: A Gentle Introduction to the Art of Programming*. Wiley & Sons, Hoboken, NJ.

Sartatzemi, M., Dagdilelis, V., Kagani, K. (2005). Teaching programming with robots: a case study on greek secondary education. *Lecture Notes in Computer Science*, 3746, 502–512.

Scardamalia, M., Bereiter, C. (2003). Knowledge building. In: *Encyclopedia of Education*, 2nd ed., 1370–1373. New York, Macmillan Reference, USA.

Shayer, M. (1997). The long-term effects of cognitive acceleration on pupils’ school achievement. Paper presented at the *Annual Meeting of the American Educational Research Association*, Chicago, Ill, March 24–28. `http://www.eric.ed.gov/contentdelivery/servlet/ERICServlet?accno=ED408195`

Solomon, C.J., Papert, S. (1976). A case study of a young child doing turtle graphics in LOGO. *AI Memo*, 375, Massachusetts Inst. of Tech., Cambridge. Artificial Intelligence Lab. `http://www.eric.ed.gov/contentdelivery/servlet/ERICServlet?accno=ED207578`

Stahl, G. (2006). *Group Cognition: Computer Support for Collaborative Knowledge Building*. MIT Press, Cambridge.

Strommen, E.F., Lincoln, B. (1992). Constructivism, technology, and the future of classroom learning. *Education and Urban Society*, 24, 466–476.

Turner, J. (1975). *Cognitive Development*. Methuen & Co. Ltd., London.

Wu, C., Tseng, I., Huang, S. (2008). Visualisation of program behaviors: physical robots versus robot simulators. In: Mittermeir, R.T., Syslo, M.M. (Eds.), *Informatics Education – Supporting Computional Thinking. Poceedings of the Third Conference on Informatics in Secondary Schools – Evolution and Perspectives*, ISSEP 2008, Torun, Poland, July 1–4, 53–62.

A. Pásztor graduated as a teacher of informatics from Eötvös Loránd University in 1993. Currently he is working towards his PhD degree at the PhD School of Computer Scients Eötvös Loránd University. His researches are how can be use programmable robots in the education of programming and the swarm-intelligence simulations with low-cost robots.

R. Pap-Szigeti graduated as a teacher of mathematics, descriptive geometry and computer science from Kossuth Lajos University in 1992. He received his PhD degree in educational science at the University of Szeged in 2009. His research is interested in development of thinking skills in primary school. He is also interested in the effects of new learning methods and tools in teaching of programming. He is a member of the European Association for Research on Learning and Instruction.

E. Lakatos Török received her MA degree in history from Eötvös Loránd University in1996. Currently she is working towards her PhD degree in educational science at the University of Szeged. Her research is focused on teachers' attitudes towards ICT and on the effects of using ICT in education. She is a member of the European Association for Research on Learning and Instruction.

Modelių robotų panaudojimo poveikis programavimo mokymui

Attila PÁSZTOR, Róbert PAP-SZIGETI, Erika LAKATOS TÖRÖK

Šiame straipsnyje nagrinėjama, kaip nauji prietaisai ir metodai gali įtakoti programavimo ugdymą. Kecskemeto koledže (Vengrija) buvo naudojami programuojami mobilūs robotai ir taikomi konstruktyvistinės pedagogikos metodai. Iškeltos hipotezės buvo patvirtintos atlikus bandymus. Priemonių naudojimu pagrįstas inovatyvus metodinis ugdymas suteikia daugiau praktinio programavimo žinių, plečią požiūrį į programavimą ir daro jį patrauklesnį. Eksperimentinės ir kontrolinės grupės rezultatai buvo palyginti pusmečio pradžioje ir pabaigoje (tuo metu buvo įvertintos programavimo žinios ir motyvacija). Per mokymosi procesą tik eksperimentinės grupės naudojosi robotų modeliais ir naujais metodais.