DE GRUYTER OPEN

ICTE Journal

2012, **1**(1): 28–39
DOI: 10.1515/ijicte-2012-0003

INQUIRY BASED LEARNING OF SELECTED COMPUTER SCIENCES CONCEPTS AND PRINCIPLES

Ľubomír Šnajder, Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University in Košice, Jesenná 5, 040 01 Košice, lubomir.snajder@upjs.sk
Ján Guniš, Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University in Košice, Jesenná 5, 040 01 Košice, jan.gunis@upjs.sk

Abstract

In the article we describe the proposal of two unplugged activities verified in practice together with methodological commentaries on how to use them in computer science education. The first activity is oriented on discovering and understanding the basic principles and rules of computer communication carried out by pupils themselves. We encourage pupils to simulate communication in a computer network and solve raised problems. The aim is to lead pupils to discover the encoded message within transmitted information (the structure of TCP/IP packet) and to realize the significance and importance of communication protocols. The second activity is oriented on discovering and devising the principles of information encoding; the code length dependence on the quantity of states, unambiguous encoding, the use of *n*-ary trees in encoding, etc.

Keywords

teaching computer science, conceptual subject matter, problem solving, experiential activities, constructionism, inquiry based learning

Introduction

The current school reform in Slovakia that launched in 2008 brought many changes to computer science teaching. Time subsidy of computer science education on the second level of elementary schools and at grammar schools has increased. Teaching the new syllabus of computer science education was introduced from the second grade of elementary schools. Not only has the school reform brought greater flexibility of choice of the teaching content to schools and to teachers; it also emphasises the use of modern teaching approaches that seek to fulfil the specific objectives of teaching computer science, and it has identified the core competencies of pupils. Schools are able to define themselves properly within the region and the teacher has greater freedom for self-realisation. From the perspective of pupil learning, the reform promises to move away from memorising towards the active involvement of pupils in the learning process.

The State Education Programme contains a framework of graduate profile, a general curriculum for each school level and a general curriculum for each school grade. Schools have created their own terms and specifics of school educational programmes based on the State Education Programme. Their implementation in practice requires adequate human and material resources (professionally and methodically skilful teachers, current textbooks, methodology and access to digital technologies). Schools and their teachers are able to acquire didactic equipment, to create their own methodology for teaching, to supplement and/or extend their education and to get thematic and methodological materials through various EU projects (ESF, MVP, DVUi).

From the viewpoint of computer science education we consider it very important to educate pre-service and in-service computer science teachers in terms of modern concepts of learning (constructivism, constructionism, connectivism). Our goal is that the teacher is able to creatively use the gained knowledge in their own teaching.

In this paper we present two well-tested activities in which we want to introduce a concept of teaching computer science where pupils actively discover the principles and procedures of computer science, similar to activities described in (http://csunplugged.org). The pupils engage their higher cognitive functions (analysis, evaluation, creativity) in learning; they are curious, thoughtful, and critical. Teachers can help to create a pleasant atmosphere of creativity (commissioning problems packed into tricks, magic, puzzles, simulations, case studies), they support the process of discovering knowledge mainly through dialogue with pupils (Socratic, heuristic, controlled) and also through their own example – when they show their thinking, their openness to the pupils' reasoning development, when they show how they correct and adapt their problem solving, etc. (Šnajder, 2009, 20010 and 2011). According to (Bell, 2008) the following two activities fall into structured inquiry and guided inquiry levels.

Activity 1: Data transmission network, IP packet

This activity is a free sequel to the activity *Data transfer in network – simulation by paperwork communication* (Šnajder, 2011) we carried out with 9 years old pupils. Its aim was to make pupils discover the basic rules of data transmission network. This next activity is more in depth and brings more of the real network communication. Pupils' activity is aimed at discovering and devising the IP packet structure and rules of communication in the Internet network.

Problem: What form and requisites should a message in the internet network have? Design the structure of this message. According to which rules governing the communication on the internet?

Thematic area: Communication through IT.

Topics: message and its parts, TCP/IP communication protocol

Input knowledge

- explain the meaning of the concept of computer network;
- propose a simple home computer network with an emphasis on functionality and safety;
- compare designs of simple computer networks in terms of safety;
- explain the function of selected network devices in a computer network.

Objectives

- specify the parts of message necessary for successful communication in the Internet network;
- explain the importance of parts of messages over the Internet network;
- explain the importance of communication protocol.

Age group: 15 years old pupils.

Time requirement: 2 lessons.

Used teaching aids and equipment: blackboard, colour markers, computer connected to the Internet network, smaller sheets of paper.

Course exercises

Pupils know the meaning of the concept of computer network, know what facilities may be located in the network and know the function of these devices.

During this activity pupils simulate a computer network and communication in it. Messages are sent in written papers. Role of the teacher is to generate an error, which can occur over real communication in a computer network. The pupils' task is to specify the rules of communication in the network to eliminate the consequences of these errors.

The teacher can generate sequences of various errors. It depends on the actions and reactions of pupils.

With pupils, we gradually discover the rules of communication and course of our activity is the following:

Identifiers of sender and recipient have to be parts of the message

We chose one of the pupils and gave them the role: "Choose one of your classmates and send them a specific question!"

Comment: The use of asking questions have been chosen deliberately. The question is naturally expected to have an answer. The pupil should be aware that it is not enough to enter the recipient (Destination IP address) of the question, but that they must also provide the sender (Source IP address) where the subsequent response to the question will be sent. We expected pupils to use sender and recipient identifiers quite naturally. However, it was not so. The pupils realised the problem only when they tried to deliver the paper sheet with answer. We assume that we have partly caused this problem too. The pupils saw each other, so everybody knew who the sender of the message was and did not consider it necessary to explicitly state their identifier.

With such an active, experiential learning, we can recognise some of the pupils' misconceptions. For example, one pupil solved this task by writing the question only on the paper. The paper sheet aimlessly "wandered" through the pupils' network and vainly searched for its recipient. When we asked the sender why he did it that way he answered that he sent SPAM. The paper with the message actually went through each pupil (even repeatedly through some). The number of the receivers was large (the same way as with SPAM), but the principles of SPAM distribution is exactly the opposite.

This situation, although we did not plan it in advance, allowed us to ask another question: *How long should this sheet of paper wander over the network?* Pupils suggested calculating the time it takes for the paper to find its recipient. If it reaches some critical value, it should be destroyed. After a series of questions such as "*Who should calculate it?*", "*How to know the time for which the paper moves between two nodes?*", "*Where should this time be recorded?*" pupils have come to the conclusion that it would be preferable to record the number of "jumps" of a paper on the same paper. Anyone who moves this paper increases this number. If the value is equal to any previously defined value, the paper is destroyed. It works similarly in computer networks (Time to Live), although there it works as a countdown.

Acknowledgment of receipt

Here, too, we asked a pupil to send a question to one of the classmates. But the teacher (the error generator) destroyed the paper on the way.

Comment: The sender sent the message, but the recipient did not know about it. The recipient did not get the message, but the sender did not know about it. How to resolve this impasse?

The pupils suggested that the recipient of the message confirmed its receipt to the sender. We repeated the communication with this new rule. This time, however, the error "occurred" during the receiving of the confirmation message. Problematic situation arose again and pupils had to solve it. They suggested that if there is no response within the time limit, the sending is repeated.

This solution is also interesting from the perspective that defines the rules for communication (communication protocol) itself. But if the rules of communication are clear in advance, is it necessary to indicate them in the message itself? If we focus on the issues of protocols, we can hold a discussion in this direction. In this activity, this rule remained part of the message itself – therefore, not as a rule of the communication protocol.

Message fragmentation

We selected a pupil and asked them to devise a way to send an extensive message to their classmate and to send it in this way.

Comment: For time reasons we did not force pupils to write an extensive message. The message can be shorter, but we agreed that only one word fits on one paper sheet. Given the intellectual maturity of the pupils we could afford this abstraction of reality, without compromising the understanding of the real problem. The selected pupil quite intuitively broke down the message to more papers and sent them to their classmate. The message reached the recipient, but it was not clear in what order the various papers were to be put together. The pupil edited their method so that he numbered each paper (Fragment Offset). The recipient therefore had no problem to compound parts of the message in the correct order. When asked whether the recipient can be sure that the message is complete, pupils said they cannot. The sender improved their method again – he added the total number of papers to each paper (like Flags). Now the recipient knows clearly how to compose the entire message and knows that it is complete.

The contents of message checking

We have asked one of the pupils to send a question to the selected classmate.

Comment: During the transmission of this message through the pupils' network we modified the question in the message. Changes were made so that the question makes sense and there is an answer to it. The recipient has sent a reply to incoming question. However, this response did not make sense in the context of the original question. The pupils' task was to find out what happened and how to prevent it.

Pupils suggested various ways for verifying the contents of the message. The message should contain e.g. a number of its letters or words. On the question of whether we can on the basis of these numbers verify with certainty the contents of the message they said we do not. Furthermore, they suggested to encode the message with 1 and 0 and to verify the number of these characters. Here they encountered the same problem. They proposed to send the message twice. If two identical messages arrive we can be almost sure the message has not been changed on its way to its recipient. The fact that we had to send twice as much data did not matter at this stage. The pupils started to think otherwise, when we asked them whether they would mind if they downloaded a film from the Internet for two hours instead of one. They had already begun to realize the consequences of such duplicity. We concluded that we cannot verify with certainty that the message has not been changed on its way to the recipient. We can, however, say it with a high probability (Header Checksum). The pupils' grasp of efficiency is also interesting. For small quantities of data we essentially do not care whether the data are sent once or twice. Pupils do not always realise the consequences of inefficiency in real situations.

Data privacy

Again we asked the selected pupil to send a question to their classmate. The answer, however, should be private. The pupil’s task was to figure out how to conceal the message.

Comment: The current method of sending the message makes it possible to read the message content for each classmate on the network. It is enough that the message on its way to the recipient passes through a particular pupil.

Pupils, based on their existing knowledge, decided to encrypt messages. They bent the paper in half to express this fact. The message itself was contained inside the paper, while the outer side of the paper contained information necessary for its delivery.

Activity evaluation

Pupils’ activity is aimed at discovering and devising the IP packet structure and rules of communication in the Internet network. Our goal is not to give the finished knowledge to pupils. Our goal is that pupils discover this knowledge by themselves and give reasons for their conclusions through solving simple problems. We are confident that many of the concepts and principles of computer science are based on common sense and logical thinking. We assume therefore, that pupils can come to this knowledge on their own (although through simplified models) under the coherent guidance of the teacher. Knowledge gained by active problem-solving fits better into students’ cognitive structure. It will have a higher durability than knowledge provided to pupils as a ready-made facts.

At first glance it may seem that the amount of time (2 lessons) dedicated to this activity is disproportionately large to achieve the objective (the structure of IP packet). Let us remember however that the activity involves the development of pupils’ skills such as competences of lifelong learning, competence in information and communication technology and competence to solve problems. Therefore the goal is not only the result itself, but also the path pupils take to understanding.

Pupils have a problem with “reducing themselves” to the intellectual level of a machine. They often transfer their own characteristics, experiences and intellect – although machines do not have them – to machine operation. E.g. they route the answer to the opposite direction than the question, do not wait for the acknowledgment of receipt of message (because I saw it), they put together received parts of a message in the order that makes sense to them and so on. At first glance it may seem counterproductive – “forcing” pupils to think so primitively; but in fact they are often confronted with the primitive level of machines. Knowing to estimate the behaviour of the machines in advance can be, in the future, very useful for pupils.

Pupils think of computers as very quick machines – moreover, performance of computers is increasing all the time. They do not realise, however, that computer is processing more data. The amount of data is growing faster than the performance of computers. We encounter this problem especially in programming. Pupils do not realise the reasons and do not feel the need to deal with the efficiency of algorithms. They are often satisfied with the fact that the program works.

At the beginning of this activity, we assumed that pupils will utilise the acquired knowledge and communication will be more sophisticated. Yet in fact, pupils always focused on solving the current problem. All paper sheets contained only a message and an address of recipient and sender. Pupils used other particular requirements only when we were solving a specific partial problem of communication.

<table>
<tr><td>Version</td><td colspan="2">IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="4">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td colspan="2">Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="6">Source IP address
Destination IP address</td></tr>
<tr><td colspan="5">IP Options</td><td>Padding</td></tr>
<tr><td colspan="6">(Enrypted) Data</td></tr>
</table>

Fig 1. The IP packet structure

We simply do not think these activities to be single-use activities for pupils. We always try to design them so that we can develop pupils' learning further on the basis of the activity results. The described activity is a suitable introduction to the layered model of TCP/IP. But furthermore, the names of the people who communicate with each other are not IP addresses. E.g. they are identified by e-mail addresses. And how the e-mail gets to the proper recipient, if the devices in the network are identified by their IP addresses? That can be the subject-matter of the next pupil activity.

Activity 2: Information encoding, binary search

Problem: Describe the way how to find out one card chosen from a package of 32 German deck cards. Do you proceed randomly or according to some rules? How many trials do you need to find the chosen card? Can you improve your procedure of finding the chosen card? Explain the relationship between finding a card and information encoding?

Thematic area: Information around us; Procedures, problem solving, algorithmic thinking.

Topics: Information encoding, calculation of information quantity, binary searching of elements in an ordered sequence according to a selected key.

Input knowledge: Operating calculations with various positional numeral systems (writing successor of a number, conversion of number notation between various bases).

Objectives

- To explain that the amount of information necessary to determine a chosen card depends on the total number of cards and also on the manner of questioning (the number of different answers to the questions).
- To explain that after each answer to a question we acquire more and more information, after binary question we acquire 1 bit of information (after ternary question 1 trit of information, after decimal question 1 dit (ban) of information).
- To give reasons that for a clear determination of one chosen card from the package of 32 cards we need to ask 5 binary questions (for one from 27 cards we need to ask 3 ternary questions, in general for determining a card from a package of K cards we need to ask $\lceil \log_N K \rceil$ N-ary questions at most).

Age group: secondary school pupils (university students).

Time requirement: 1 lesson.

Used teaching aids and equipment: a package of 32 German cards.

The progress of solving the problem

In the first phase of solving the problem the pupils themselves play the game in pairs, where they alternate in guessing the chosen card. In the second phase, by means of the Socratic debate with teacher, pupils discover a sophisticated way of determining the chosen card. Finally, in the third phase after playing the card guessing game, pupils study information encoding using binary numbers, calculating the amount of information, using binary tree for searching, information encoding and showing the uniqueness of this encoding.

In the first motivational and experimental phase teacher motivates pupils to discover the best way how to guess a card chosen by a classmate by themselves. Before pupils' experimentation, it is important for teacher to warm up their thinking by asking some questions, for example. "Do you know the package of 32 German cards? What kind of suits (colours) and values are there? How many cards are in the whole package?" Clever pupils can carry out the activity without manipulating the tangible objects – cards, using only the help of their imagination and thinking. We encourage pupils to play the game several times in different roles and also to record number of questions for each card guessing. The first phase we conclude with questions, e. g. "How many attempts do you need during the card guessing? Who was better? Did you guess randomly, or did you use a sophisticated way of finding out the card?"

In the second phase of solving the problem we use the experience acquired by pupils from the first phase and by Socrates debate we try to analyze pupils' random guessing and jointly come to a systematic way of finding the card – the binary search algorithm. An example of possible dialogue:

Teacher: Considering the random guessing of the card, can we guess the card on the first try?

Pupil: Yes, but in this way we will not always be successful.

Teacher: How many guesses can we reach in the worst case?

Pupil: Well, if we have a really bad luck, then 32 guesses.

Pupil2: We can maybe make more than 32 attempts, but I think that 31 attempts will be sufficient for guessing it in the worst case.

Teacher: What can we say about a set of cards with the desired card during gradual questioning?

Pupil: That this set will be reduced question by question.

Teacher: How will it be reduced?

Pupil: After each question this set will be smaller by one card.

Teacher: Yes, in a random guessing a set with the desired card will shrink by 1 until we guess the right card.

Next, the teacher puts the cards on the table face up and organizes them by their colour (suit) and values. They can also use a projector for showing the cards to all pupils and continue with the dialogue.

Teacher: Try to ask a question in order to reduce the set with the searched card by more than only one card

Pupil: Is it green?

Teacher: Yes, great. How big will the set be afterwards?

Pupil: If the answer is 'yes' then the desired card will be one of 8, otherwise one of 24 cards.

Teacher: OK. Try asking a question in a way that regardless of the answer, you will have equal chances of finding the card in either group.

Pupil: Is the value of the searched card a number?

Teacher: Excellent. And how will it reduce the set with the desired card?

Pupil: By half.

Teacher: Yes, exactly. Now try to ask a question concerning the colour of the desired card that will narrow the set in half.

Pupil: Is it a red card or green?

Teacher: Excellent. Is the colour of the searched-for card one of the following colours: red, green? What will be the further question concerning the colour of the searched-for card?

Pupil: Well, after this question I know that the desired card is either from the colour set {red, green}, or colour set {acorn, ball}. Then I ask for the one colour from the selected set. E. g. when the first set is chosen, so I ask – Is it red? And then I definitely know the colour of the desired card.

Teacher: Yes, first you asked about one pair of colours and then you asked about one from the pair. At the beginning we have 32 cards. After the first "yes/no" answer the set with the searched-for card is reduced to 16 elements, after next answer to 8 elements. I believe that it wouldn't be difficult for you to propose further questions for determining the value of the searched card.

Pupil: I ask whether the card is a number. After answering this question only 4 cards remain. Either they will be numbers, or figures. After my next question only 2 cards will remain and finally by the last question I will definitely know the desired card.

Teacher: Perfect. While asking the questions it is important to find and formulate a convenient attribute through which a set with the desired card can be reduced. How many questions did we need for guessing, or finding the searched-for card?

Pupil: 5.

Fig 2. Gradual narrowing of a set with the selected card from 32 cards to 1 card

Teacher: Yes, we need 5 questions. At the beginning the set with the desired card has 32 elements. After each answer this set is reduced by half gradually to 16, 8, 4, 2 cards and finally to 1 card, which is illustrated in Fig. 2.

After pupils discover the way of questioning, the teacher leads a dialogue to generalise the solution of the problem. Pupils can very easily discover the fact that for finding a card from 64 cards it is necessary to ask 6 questions, and from 1024 cards 10 questions are needed. Then the teacher shows that for $N=2^K$ cards the sequence of cards with searched card is decreasing as follows: 2^{K-1}, 2^{K-2} ... $2^{K-K} = 2^0 = 1$. Pupils can conclude from this sequence that for finding a card from package of 2^K cards it is necessary to ask K "yes/no" questions. At secondary school level pupils who know logarithmic function can express that for finding a card from N cards it is necessary to ask $\log_2 N$ questions. They can also discover a formula for $N \neq 2^K$ cards.

There is a binary tree that is considered a useful tool for understanding the relationship between the total number of elements (e.g. cards) and the number of bits necessary for encoding all the elements and also the number of binary questions for finding out a searched-for element.

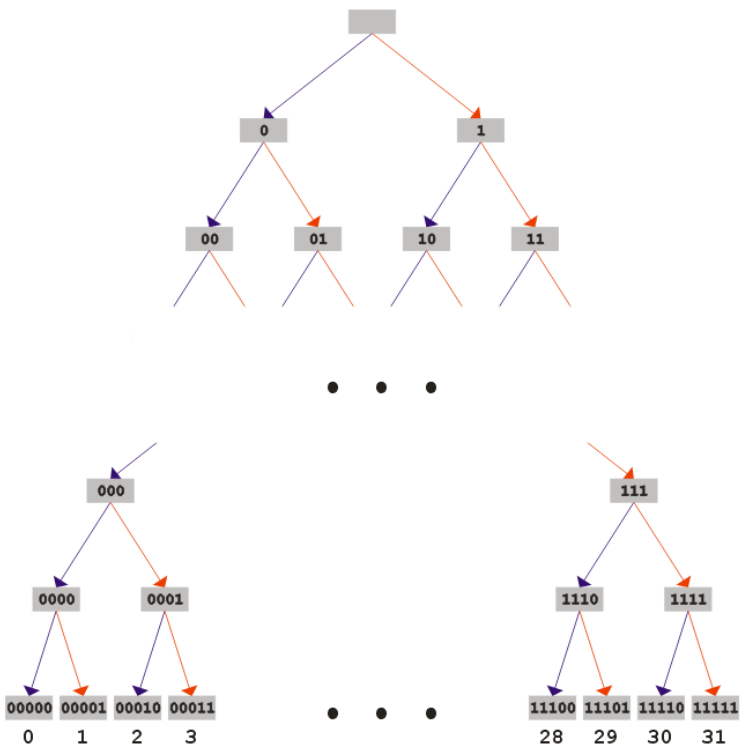

Fig 3. Binary tree with the depth of 5 and 25 = 32 leaves assigned by binary numbers

Let's create a binary tree (Fig. 3) starting with an empty root node. From the node we draw two edges leading to two child nodes. Let's assign the left child node with the value from the parent node and add a "0" digit to the right side of that value, and the right child node with the value from the root node adding a "1" digit to its right side. This procedure can be applied for each node. It is clear that in K^{th} depth of binary tree there are vertices that are assigned by numbers with K binary digits. If we would like to find an arbitrary number with K binary digits we need to know answers for K binary questions. In other words we can say that for the representation of a number from 2^K numbers we need K binary digits or acquire K bits of information. In the guessing card game we can receive one 5-bit information (e.g. red king) to know definitely all about the searched card, or one 2-bit information about its colour (e. g. red) and one 3-bit information about its value (e. g. king), … or five 1-bit information illustrated in Fig. 2. Previous consideration shows relation between bit as a binary digit and also as a binary information unit.

From the binary numbers we can move to N-ary numbers. For guessing, or finding one of 2^{10} numbers we need to ask 10 binary questions and to acquire 10 bits of information. For finding out one of 10^3 numbers we need to ask 3 decimal questions (value of 1st decimal digit, 2nd decimal digit, and 3rd decimal digit) and to acquire 3 bits of information. For determining the chosen card among 3^4 cards organized in square with 9×9 positions, we need to ask 2 nonary questions or 4 ternary questions. In general, for determining one of the K^L elements we need to acquire L K-nary information. Examining a K-nary tree, pupils can deduce that K-nary tree with the depth L has K^L nodes, and also that the depth of K-nary tree is closely related to logarithmic function (with the base K) and number of digits of a K-nary number.

Comments, alternatives, enhancements, advices

This methodology was based on the experience of teaching in different target groups (from 3rd grade elementary school pupils, gifted pupils from 7th to 9th grades of elementary school, the 1st year of secondary school pupils, to pre-service and in-service computer science teachers) and on different tasks; searching words in a dictionary, guessing integer numbers from 1 to 16, guessing one of the 32 (or 27) cards obtaining binary (or ternary) answers, etc.

Of course for different target groups we have achieved different range and depth of subject matter comprehension. Pupils from 3rd grade of elementary school have discovered for themselves a better way of finding a word in the dictionary – utilising the alphabet ordering of words in the dictionary and remembering boundaries of pages where the chosen word can be found and then browsing the dictionary page by page. They understand that "spider" (binary tree) is a good aid for guessing integer numbers (Šnajder, 2011).

Gifted pupils from 7th to 9th grades of elementary school very quickly discovered a principle of effectively finding a card using binary (or ternary) questions. They were very active, asked many questions, created and verified hypotheses, and were very consistent and critical (they verified the card package in order to ensure that it is not an irregular one). Only some of them knew how to work with other than decimal numbers, but after using the above described Socratic debate and experimenting, all of them understood the principle of representing numbers in other than decimal system.

Pre-service and in-service computer science teachers appreciated the magic trick with cards as a very appropriate activating teaching method which helps to better understand selected computer science principles and they also joined the solving with great interest and pleasure. During a lecture with a group of pre-service computer science teachers a lively discussion arose. Students replied several times to the trick of guessing one out of 27 and more cards. They were surprised to find the searched-for card from a package of 30 cards after 3 ternary answers. After explanation of this issue with regards to probability students have comprehended the whole problem.

Teachers can use a special set of cards instead of German cards, e.g. cards with computer devices, binary numbers, animals, fruit, etc. that could be more appropriate for younger pupils but also to remember the selected notions of computer science and for closer relation to other school subjects.

Although we did not use computer support in our methodology we think that computer animations, interactive applets etc. could be helpful for individual discovery of the selected notions, principles and relations and also for their better illustration and revision. On the other hand we consider the role of the teacher to be crucial in the methodology because of their creativity and ability to lead adaptive dialogues with pupils.

To achieve satisfying results of learning regardless of target groups' diversity we recommend heeding the following aspects:

- To motivate pupils to solve a given problem (e.g. how to reveal the principle of the magic trick), to encourage them by appreciative words.
- To maintain pupils' activity by asking questions, discussing and experimenting that help to develop pupils' creativity and critical thinking.
- To provide adequate amount of questions and clues/hints and to do it in proper order so that pupils may discover new knowledge by themselves or with little help by the teacher.
- To repeat an idea several times from various points of view in order to clarify and consolidate new knowledge.

Conclusion

In this paper we have mentioned tasks aimed at understanding some concepts and principles of computer science. The tasks are directly applicable to the lessons using various levels of inquiry based learning with regard to pupils' age or inquiry experience. We are not focusing only on what a pupil can do alone. What pupil can do today, even with the help of the teacher, they will be able to do alone tomorrow (Vygotsky, 2004). So a good learning precedes pupil's current development level. Although the current development level of pupils (their level of knowledge) may be on the same or very similar levels, not every one of them will be able to move forward at the same pace; the zone of proximal development of each of them may be different. Poor consolidation of knowledge seems to be a "weakness" in the Socratic method (learning through questions). Instead of breadth we focus on depth in learning and the proper understanding. Instead of specific procedures and tools we prefer reasoning, contemplation, curiosity and critical thinking. The method of questioning is also important. Questions should be asked logically, not psychologically (Socratic method, 2011, online).

The findings, published in this paper, were developed under a grant LPP-APVV 0057-09 and grant KEGA 326-009UPJŠ-4/2010.

References

BELL, R. The Many Levels of Inquiry. *Science and Children* [Online]. 2008, 46, 26-29. [Accessed 2011-12-15] Available at: http://learningcenter.nsta.org/files/sc0810_26.pdf.

Computer Science Unplugged: Computer Science Unplugged [Online]. 2011. [Accessed 2011-12-14]. Available at: http://csunplugged.org/.

GARLIKOV, R. *The Socratic Method*. [Online] 2006. [Accessed 2011-12-09]. Available at: http://www.garlikov.com/Soc_Meth.html.

KALAŠ, I. et al. *Východiská a inšpirácie, Línia: Moderná škola. Ďalšie vzdelávanie učiteľov základných a stredných škôl v predmete informatika*. Bratislava: Štátny Pedagogický Ústav, 2009. ISBN 978-80-89225-51-4.

ŠNAJDER, Ľ. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Metodika výučby tematickej oblasti Informácie okolo nás*. Bratislava: Štátny pedagogický ústav, 2010. ISBN 978-80-8118-051-4.

ŠNAJDER, Ľ. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Aktivizujúce metódy vo výučbe školskej informatiky*. Bratislava: Štátny pedagogický ústav, 2009, ISBN 978-80-89225-96-5.

ŠNAJDER, Ľ. Úlohy zamerané na aktívne učenia sa vybraných pojmov a princípov informatiky. In: *Didinfo 2011*. Banská Bystrica: UMB Banská Bystrica, 2011.

VYGOTSKY, L. S. *Psychologie myšlení a řeči*. Praha: Portál, 2004. ISBN 80-7178-943-7.