PROCEEDINGS

of the Union of Scientists - Ruse

Book 5

Mathematics, Informatics and Physics

Volume 8, 2011

RUSE

ISSN 1311-9974

The Ruse Branch of the Union of Scientists in Bulgaria was founded in 1956. Its first Chairman was Prof. Stoyan Petrov. He was followed by Prof. Trifon Georgiev, Prof. Kolyo Vasilev, Prof. Georgi Popov, Prof. Mityo Kanev, Assoc. Prof. Boris Borisov, Prof. Emil Marinov. The individual members number nearly 300 recognized scientists from Ruse, organized in 13 scientific sections. There are several collective members too – organizations and companies from Ruse, known for their success in the field of science and higher education, or their applied research activities. The activities of the Union of Scientists – Ruse are numerous: scientific, educational and other humanitarian events directly related to hot issues in the development of Ruse region, including its infrastructure, environment, history and future development; commitment to the development of the scientific organizations in Ruse, the professional development and growth of the scientists and the protection of their individual rights.

The Union of Scientists – Ruse (US – Ruse) organizes publishing of scientific and popular informative literature, and since 1998 – the “Proceedings of the Union of Scientists- Ruse".

BOOK 5

"MATHEMATICS, INFORMATICS AND PHYSICS"

VOLUME 8

CONTENTS

Mathematics

Informatics

Physics

BOOK 5

"MATHEMATICS, INFORMATICS AND PHYSICS"

VOLUME 8

Education

Participation in International Projects

AN INNOVATIVE APPROACH TO INFORMATICS TRAINING FOR CHILDREN

Plamenka Hristova [1], Neli Maneva [2]

[1] Angel Kanchev University of Ruse, Informatics and Information Technology Department
[2]Institute of Mathematics and Informatics, BAS, Software Engineering Department

Abstract: *The paper presents our attempt to construct an innovative approach to Informatics teaching, which has been designed especially for pupils up to 5-th grade. The rationale and main ideas of the approach, based on Comparative analysis, are briefly described. An example illustrates the feasibility and usefulness of the proposed method for a given real-life situation. Finally, some intentions for further research and experimental work are shared.*
Keywords: *Extra-curricular Informatics teaching, Competitive programming, Multi-criteria decision making, Comparative analysis*

INTRODUCTION

European politicians have recognised that education and training are essential to the development of today's knowledge society and economy. The EU's “ET2020” strategy states the main goal of school education – to reflect the changing needs, equipping the new generation with useful basic knowledge, skills and competencies, together with properly defined and developed fundamental attitudes and values [8]. In this strategy Informatics and Information technologies are among the areas, mentioned as key ones.

Unfortunately, for lower secondary schools in Bulgaria (and in most European countries), Informatics is not a compulsory subject. There are no officially published and approved curriculum and educational standards. In order to make possible training of pupils at this age, the Bulgarian National commission for extracurricular activities on Informatics created a curriculum, comprising 15 themes. These themes are studied only in different out-of-class courses, comprising lectures and practical exercises. Students, attending such courses, have to learn a number of basic algorithms and how to write programs for them in a given programming language, mostly C++. But defining the scope and content of curriculum is only the first step. Especially for instructors of pupils up to 5-th grade, Informatics teaching is still a great challenge, because of the lack of well developed and validated pedagogical approaches and methods, ensuring smooth and efficient training process.

The current paper presents our vision about a systematic approach to Informatics training of children. Part 2 introduces the rationale and the basic ideas on which the approach has been built. Part 2.1 describes briefly the method of Comparative analysis, and Part 2.2 - its possible use for Informatics teaching. In Conclusion some ideas for further research and practical work have been given.

AN APPROACH TO INFORMATICS TEACHING

On the base of analysis of the current state and our experience gained in the field, we reach the understanding that the above mentioned directives for school education in Informatics can be accomplished successfully through a new, non-standard approach. Searching for some science-based and innovative solutions, we decided to introduce the so-called CCC approach, meant to be:

- Constant – to apply a systematic rather than ad-hoc approach, following a consistent and long-term strategy;

- Continuous – to start with some procedural regulations and their use for a few selected activities in teaching and only after their successful adoption to move to other ones;
- Correct – based on some already validated methods, techniques and best practices.

From pragmatic point of view, the first and the second principles cover some managerial aspects and describe ***how*** we have to apply a number of appropriate methods. So the crucial factor for the success of this approach is ***the choice*** of methods to be used - in accordance with the third principle and taking into account the specifics of Informatics training for children in this age group.

Next follows a brief description of one universal and validated method – the Comparative analysis. We are going to show how it can be used, independently or through integration with another method, thus forming a pair of mutually complementary and compatible ones.

1. Comparative Analysis

Generally speaking, the Comparative analysis (CA) shares the main objectives and methods of the theory of multiple criteria decision making [1], but in expanded form so as to specify and use them systematically in a selected application area.

The detailed description of CA can be found in [3]. Here we want to present only a piece of information, necessary for understanding the CA method usage.

The method of CA is a study of the quality content of a set of homogeneous objects and their mutual comparison so as to select the best, to rank them or to classify each object to one of the predefined quality categories.

At the beginning of CA use, we have to identify objects in the selected application area, worth to be studied. ***Object*** can be any item under consideration. Being in a given state, the object posses specific q*uality content*, described and evaluated by its *model*. The modelling goal is to obtain a *quantitative measure of object quality* on the base of some user-oriented quality factors through their decomposition into measurable characteristics in one or more levels.

Two main players participate in the process of CA usage – Analyst and Customer. The ***Analyst*** is responsible for models construction and for accomplishing the whole CA procedure, developed for the problem, which has been defined by the Customer. Depending on the Customer's role, the stated problem and the current moment, a ***case*** should be opened to determine the *context* of the desired comparative analysis, described by the following six elements:

case = {View, Goal, Object, Competitors, Task, Level}

The ***View*** describes the Customer's role and the perspective from which the comparative analysis will be performed. The ***Goal*** can be to *characterize*, to *evaluate*, to *predict* and *any other* – defined by the Customer. The ***Object*** represents the item under consideration. According to the stated Goal, the instances of the objects to be compared are included in the set C of ***Competitors***. The CA ***Task*** can be *Selection* (to find the best), *Ranking* (to obtain a completely ordered list), *Classification* (to define the appropriate quality category for each object) or *any combination of them*. The depth ***Level*** defines the overall *complexity* of the CA and depends on the importance of the problem under consideration and on the resources needed for its implementation.

2. CA application for Informatics teaching

In order to apply the CA in this area some basic objects (trainee, syllabus, topic and problem) and a number of additional objects (instructor, contest, parent, textbook, learning material, external evaluator, etc.) have been identified in [2]. Next we will describe briefly only the derivative model for the object 'trainee', most suitable for our current research goals.

Trainee is a student up to 5-th grade, attending an extracurricular course so as to acquire knowledge and skills necessary to be successful in Competitive programming.

In order to create the trainee model, we have to identify some pupil's individual characteristics, which are worth to be evaluated and, if possible, systematically developed. The basic considerations about student's profile for Informatics training have been described in detail in [2] and are shown in Fig.1

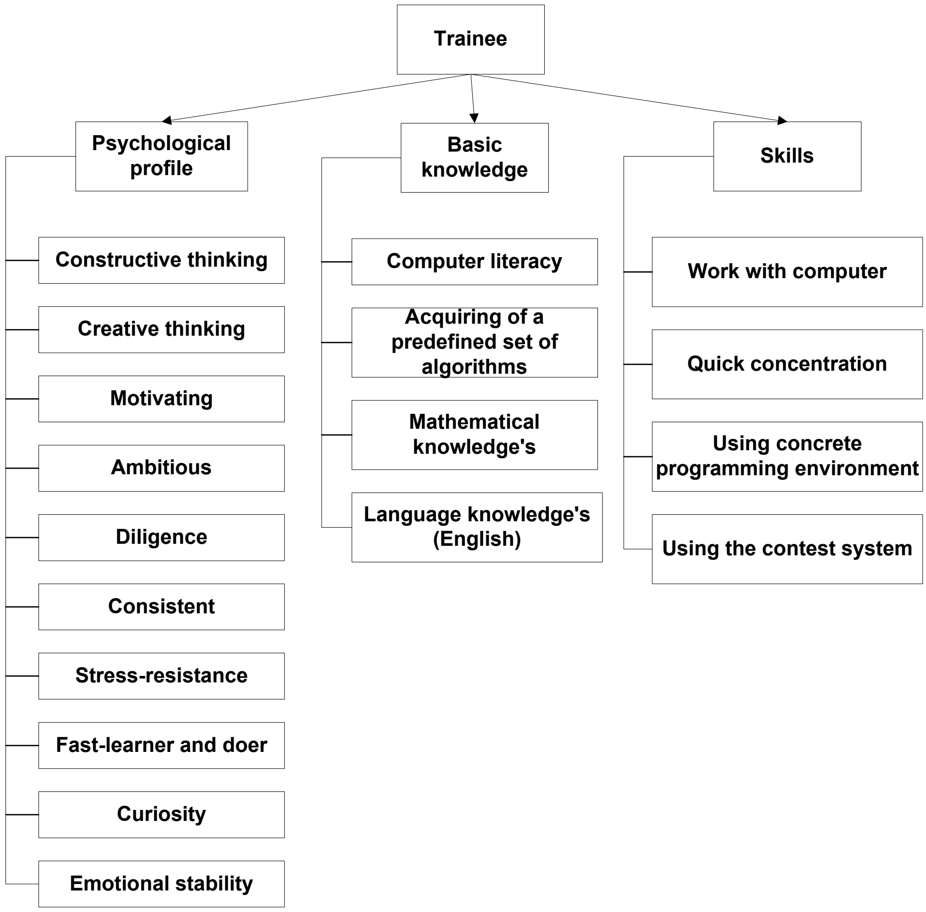

Fig.1 A model for object "Trainee"

For any defined case, involving the object Trainee, we can construct a derivative model, comprising the distinguishing trainee's characteristics, which are valuable for that case.

As an example, let's present the quality characteristic ***motivation*** from pedagogical perspective, describing its significance, influencing factors and ways for increasing its role.

From psychological point of view, the *motivation* assures the power for well organized, goal-oriented teaching. A motivated child accepts easier the burden of acquisition of different in volume and complexity information, which should be further processed - understood and memorized. A few factors can influence the motivation. The external factors can be a short-term (participation and good performance at incoming competition) or long-term – preparation for the future professional career in the field of software industry. An internal factor can be curiosity and search for intellectual satisfaction from the process of revealing the secrets of Informatics and programming. Our experience and discussions with colleagues show, that for the pupils at this age the motivating factors have more narrow scope and are connected with some concrete goals, like:

- to participate and to be a winner in a competition;
- to learn how to work in a given programming environment;
- to become a proficient in algorithm construction and writing nice programs for it;
- to develop an aesthetic approach to programming, based on smart and sophisticated tricks, search for optimized solutions, etc.

Many participants in such extracurricular forms manage to preserve the created fascination from programming during their whole life.

Therefore, the instructor should try to find which motivating factors are valid for a particular student and to make an attempt to use them properly, in accordance with the defined teaching objectives at the moment. Creating and maintaining the proper level of motivation is very important especially when a new knowledge has to be acquired. The lecturing plan should be carefully designed so as all consecutive steps (provoking the interest, introduction, explanation, examples of usage, etc.) to be performed in a proper and effective way.

Beside the motivation, many other trainee's characteristics can be defined as ones, worth to be evaluated and developed on purpose during the process of training, namely *competitiveness*, *stress-resistance*, *curiosity*, *flexibility*, *emotional stability, self discipline, risk-taking, reflectiveness, hard-working, etc*.

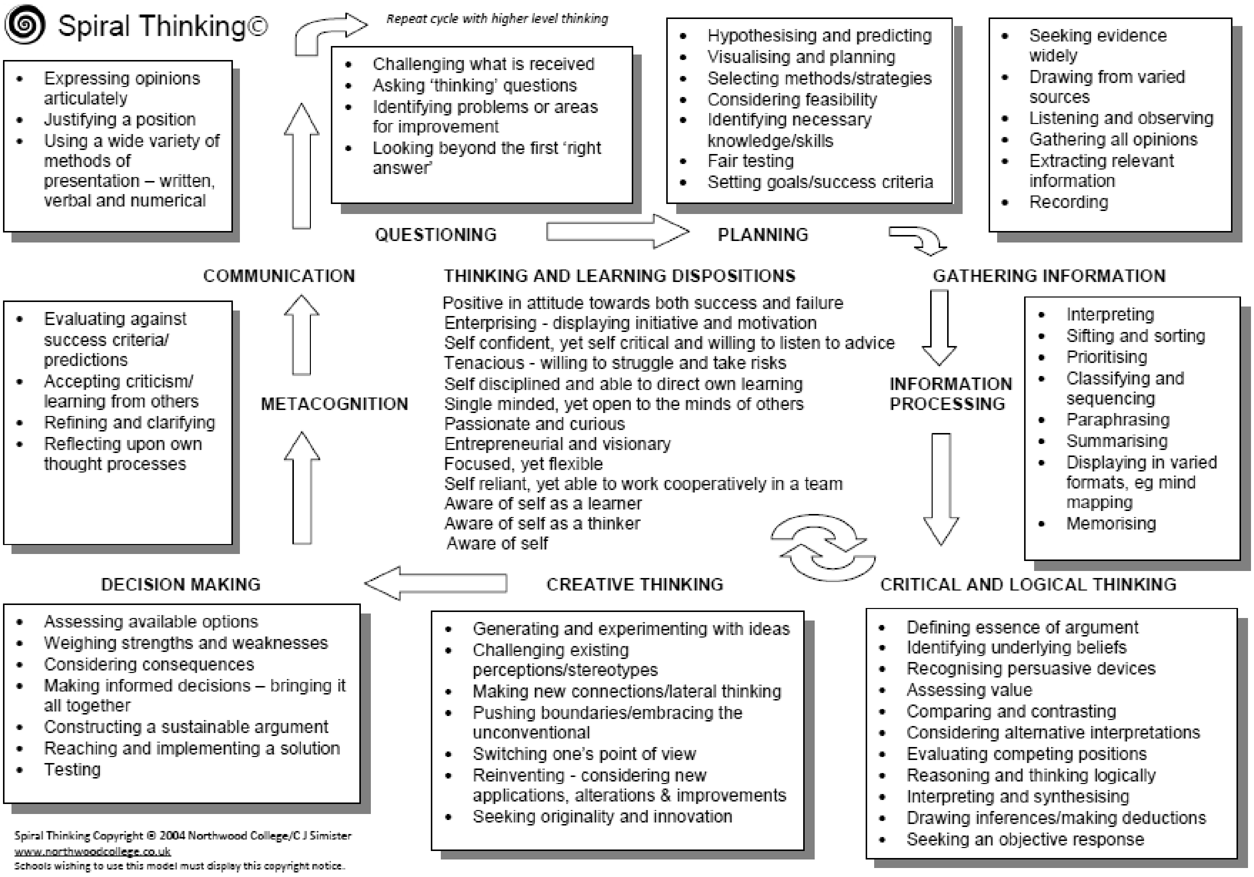

Fig.2 Spiral Thinking ©2004: C. J. Simister/Northwood College

From methodological point of view, really excellent results can be achieved by unifying the efforts of teachers with different background. The Informatics teachers can adopt a few techniques and methods, proposed especially for children in this age group and developing some significant for programming abilities. For example, the Comparative analysis, introduced here, can be applied together with the method of Spiral Thinking (see Fig.2). Our impression is that the CA can support most of the basic activities, identified as significant in this approach.

There is a number of other Thinking and Learning Skills Teaching approaches, developed for children up to 14 years [5,6]. On the base of teaching materials from these books, many important for Informatics students' quality factors can be further decomposed so as to facilitate their objective evaluation within the CA. Adopting the Spiral Thinking method [5], we can receive a description of some main abilities through a number of sub-abilities and skills, as shown in the Table 1.

Factor: Critical and logical thinking	**Factor: Creativity**
• Defining essence of argument • Identifying underlying beliefs • Recognizing persuasive devices • Assessing value • Comparing and contrasting • Considering alternative interpretations • Evaluating competing positions • Reasoning and thinking logically • Interpreting and synthesizing • Drawing inferences/making deductions • Seeking an objective response	• Generating and experimenting with ideas • Challenging existing perceptions/stereotypes • Making new connections/lateral thinking • Pushing boundaries/embracing the unconventional • Switching one's point of view • Reinventing - considering new applications, alterations and improvements • Seeking originality and innovation

Table 1. Main abilities through a number of sub-abilities and skills

Different cases can be defined in order to support the decision making in some crucial for the training moments. Each case should be constructed by the joint efforts of the Analyst and a Customer, who can be an instructor, a parent, the principal of school, etc. The analysed situation can be related with an event (e.g. internal or external competition, final exam, programming duel) or with a specific didactic goal. For example, a case can be opened so as to evaluate the quality of a written program on the base of a set ot user-defined criteria (reliability, correctness, maintainability) or/and on the base of compliance with some recommendations for good programming style.

After a proper CA implementation, the obtained CA results can be used for accomplishing the very modern nowadays "participatory" education as:

- inventing different stimuli to increase student's motivation;
- keeping a personal record for achievements and defining some individual take-home assignments;
- rescheduling the syllabus, changing not only the content and time devoted to the basic themes, but also the didactic methods, used to present them.

3 Example

In order to prove the feasibility and usefulness of the CA, let's consider a real-life situation, for which the proposed in [3] procedure will be accomplished. Next follows a brief description of this step-wise procedure, demonstrating how we can apply the CA for a defined situation.

Step 1. ***Pre-analysis*** – description of a problem, for which the CA method seems useful.

During the teaching process a number of milestones can be defined, when an evaluation of the overall trainee's performance should be made. For example, before a contest (regional or national) the instructor can organize an internal qualification test so as to select the participants, representing the group.

Step 2. ***Preparation*** – the elements of the defined case are:

View – instructor's;

Goal – assessment of the current students' level so as to rank them according to the acquired knowledge, proved skills, and demonstrated performance;

Object – a trainee, enrolled in a programming course;

Competitors – other trainees from the same group and one "virtual" trainee, presented by threshold values of the quality characteristics under consideration.

Task – ranking the trainees;

Level – high.

Step 3. Construction - building an object model, corresponding to the quality content appropriate for the situation.

For that case we choose characteristics, split into two groups: psychological traits and programming knowledge and skills, namely:

A) Psychological characteristics

F1 – motivation, $w_1 = 0,13$;

F2 – constructive thinking, $w_2 = 0,12$;

F3 – stress-resistance, $w_3 = 0,13$;

F4 – the ability for fast concentration and keeping its high level during the whole contest, $w_4 = 0,12$.

Б) Programming knowledge and skills

F5 – theoretical knowledge, $w_5 = 0,15$;

F6 – ability to apply the known algorithms for solving a new problem, $w_6 = 0,18$;

F7 – ability to understand and realize the essence of a problem with a long and not completely formalized description, $w_7 = 0,05$;

F8 - prerequisite knowledge in Mathematics, $w_8 = 0,05$;

F9 – additional knowledge about programming environments and operating systems, and skills to use them, $w_9 = 0,07$.

The assigned weights are exemplary and can be changed, depending on the context – e.g. the moment of evaluation, the type of the incoming competition, the average performance of the group till now, etc.

Step 4. Execution - Carrying out the CA task

We estimate the current overall "shape" of a trainee so as to determine his/her rank.

Trainee/ factors	**F1 (0,13)**	**F2 (0,12)**	**F3 (0,13)**	**F4 (0,12)**	**F5 (0,15)**	**F6 (0,18)**	**F7 (0,05)**	**F8 (0,05)**	**F9 (0,07)**	**Total**
virtual	2	2,5	1,5	1	2,5	1,5	1,6	1	1	***1,72***
1	2,5	1	2	1,5	2,25	2	1,3	0,5	0,7	***1,721***
2	3	2	1	1	2,5	2	1	1	0,5	***1,75***
3	1	0,5	0	0,5	2	0,5	1	1	2	***0,88***
4	0,5	0	2	0,5	1,7	1	1,4	0,5	1	***0,985***
5	0	1	0,3	0,3	1,8	4	1,8	1	2	***1,465***

Table 2. Models results after implementation a test

After each evaluation (by test, quiz, take-home assignment, contest, etc.) the results are saved in a table with rows for trainees and columns for factors). The first row comprises data about the “virtual” student with some appropriately defined threshold values. After this step, the results for a group of 5 students are shown in Table 2.

Step 5. Completion - analysis, interpretation and drawing conclusions from the results.

Trainee	virtual	2	1	5	4	3
Total score	*1,72*	*1,75*	*1,721*	*1,465*	*0,985*	*0,88*

After sorting by the Total score, the trainees will be ranked in the following way:

I – trainee 2, II – trainee 1, III – trainee 5, IV – trainee 4, V – trainee 3.

Some conclusions about the current “shape” of each trainee can be made so as to make possible planning some future instructor’s actions, like selecting the best participants for incoming contest, extra training on some stumbling blocks, assigning additional tasks for trainees with poor results and so on.

CONCLUSION

According to the research results, which have been obtained till now, the proposed innovative approach seems to be general-purpose, flexible enough to reflect the peculiarities of different context of Informatics training for children and last, but not least, feasible.

Future research work in this area will be to expand the existing object models both in variety and scope. The complete application of the approach is planned for extracurricular training on Informatics for the next academic year. We hope that such real-life experiments will enrich the set of useful principles, procedures and CA cases, thus improving the approach as a whole.

REFERENCES

[1] Dyer J., MAUT-Multiatribute Utility Theory. In: Multiple Criteria Decision Analysis: State of the Art Surveys. J. Figueira, S. Greco, M. Ehrgott, Eds. London, Springer Verlag, 2004, pp. 265-297.

[2] Hristova P., Object-Based Approach to the Extracurricular Activities in Informatics for Pupils up to 5-th Grade. PhD Thesis (in Bulgarian), Russe, 2010.

[3] Maneva N., INSPIRE: A Software Engineering Methodology. Journal of Information Technologies and Control, № 1, 2005, pp.31-37.

[4] Reis S., Small M., Gifted and Talented Learners: Many, Varied, Unique and Diverse, Conn., The National Centre of the Gifted and Talented, 1995

[5] Simister J., How to Teach Thinking and Learning Skills: A Practical Programme for the Whole School. Paul Chapman Publishing, 2007

[6] Semester J., The Bright Stuff: Playful ways to nurture your child’s extraordinary mind. Pearson Education Ltd, 2009

[7] Verhoeff T., The role of Competitions in Education, http://olympiads.win.tue.nl /ioi/ioi97/ffutwrld/competit.html.

[8] www.eurunion.org/eu/images/stories/eufactsh-eur2020-8-10.pdf (last visited on April, 30, 2011)

CONTACT ADDRESSES
Plamenka Todorova Hristova, PhD, Assist. Prof.
Angel Kanchev University of Ruse, Department IIT
tel. +359 82 888 326
e-mail: ptx@ami.uni-ruse.bg

Neli Miltcheva Maneva, Assoc. Prof.
Software Engineering Dept., Institute of Mathematics and Informatics – BAS
1113 Sofia, "Acad. G. Bonchev" str. bl.8
tel. (02) 979 28-75
e-mail: neman@gbg.bg

ЕДИН НОВАТОРСКИ ПОДХОД КЪМ ОБУЧЕНИЕТО ПО ИНФОРМАТИКА НА ДЕЦА

Пламенка Христова, Нели Манева

Русенски университет „Ангел Кънчев", ИМИ – БАН

Резюме: *Статията представя опита ни да конструираме иновативен подход към обучението по информатика, предназначен за деца до 5 клас. Основанията ни за разработване и ключовите идеи на подхода, базиран на Сравнителен анализ, са представени накратко. Приведен е пример, илюстриращ осъществимостта и полезността на предлагания метод за дадена реална ситуация. Накрая са споделени някои намерения на авторите за по-нататъшна изследователска и експериментална работа.*

Ключови думи: *Извънкласно обучение по Информатика, състезателно програмиране, многокритериално вземане на решения, сравнителен анализ*

ISSN 1311-9974