# SMPR TRAINING-METHODOLOGICAL SYSTEM FOR SUPPORT THE TRAINING COURSES ON DECISION-MAKING THEORY

**Oleksii Voloshyn, Daniil Kovaliov**

***Abstract***: *The software for support the training courses on decision-making theory in the form of an educational, methodological and evaluating system is considered. The systems purposes, its design, structure and functionalities are described. The article turns attention to the latest extensions of the system, such as the rules of fuzzy inference during the process of complex user's knowledge estimation. The perspectives of future work are also discussed.*

***Keywords:*** *protection system, e-learning, decision-making theory, evaluation, team development.*

***ACM Classification Keywords***: *D.2.13 Reusable Software - Reuse models.*

## Introduction

The SMPR software system for support the training course "Systems and Methods for Decision Making", which is taught at the Cybernetics Department of the Kyiv Taras Shevchenko University for 3-year students majoring in "Computer Science" and corresponds to training aids [Voloshyn, 2001, 2006, 2010], is represented in the report.

The software development had been started in 2005-2006, the first version of SMPR-06 software system, presented at the conference MeL-2006 and published in [Voloshyn, 2007], was designed as a set of separate software modules implementing some algorithms for solving decision-making theory (DMT) tasks according to [Voloshyn, 2001, 2006]. Software modules (developed by students at laboratory classes) operated independently, were implemented in different programming languages and had arbitrary structure and visual interfaces.

Version of SMPR-08 system [Voloshyn, 2010] provided the opportunity of modules interaction through forming of uniform requirements to their software implementation realized in programming language C # using platform .NET, development frameworks MS Visual Studio and technology for team development SVN [Gamma, 2006]. More than 50 students had participated in the development of all SMPR versions. SMPR-08 system can be called "training-methodological and demonstration-testing" due its functional content. Along with the demonstration of a specific task solution DMT user could verify the solution of a task and getting a grade "correct" or "incorrect".

Student knowledge assessment function is realized in SMPR-10 (represented at MeL-2010, Kiev, 2010, published in [Voloshyn, 2012]), the system's interface is enhanced, particularly in the using of description languages – Ukrainian, Russian, English and Chinese (individual modules). SMPR-10, as the application for training aid [Voloshyn, 2010], has been used in a number of higher educational establishments of Ukraine (in particular, Kiev, Odessa, Uzhgorod, Cherkasy, Chernihiv, and others (e.g., [Bondarenko, 2011, 2012], [Solomitskyi, 2012; Gaivoronskaya, 2013]) as support for training courses in decision-making theory, using training aids [Voloshyn, 2001, 2006, 2010].

SMPR-12 version, represented in the report, can be considered as the "commercial" version of the system. Compared to SMPR-10, two main functions was added to SMPR-12: 1) student's knowledge assessment on the entire DMT course, output in “clear” or “fuzzy” form [Voloshyn, 2010]; 2) second function is related to the protection of the system against unauthorized usage.

The authors have planned to modernize the system in the next SMPR version, by transferring it to the Internet environment; they are also planning to publish the training aid in Russian and English (and in the future - in Chinese). Ideally, SMPR software should be a part of online educational project, which is being developed in the walls of the Kiev Taras Shevchenko University, like the world-famous projects courser.org, udacity.com etc.

## General information about SMPR software system

System consists of a core and a set of specialized modules, which are responsible for solution of certain class tasks. The core creates modules functioning environment with ability of solving tasks in parallel and sharing data between modules. It consists of the common interfaces, data exchange standards, boost and information systems about module.

The core comprises a buffer that is the intercommunicator between modules. Buffer can manipulate various source data, as well as the results of their processing. It is the "environment" of the system-level data, i.e. data that can be accessed by any module. At the architectural level, the buffer is represented as a special class providing the interface for data download, store and validate for interior modules. The core of the system is not specified for specific algorithms of a particular course. Specifics of a task are provided only by the set of exterior modules, which is available to be used by the core. This facility of the system core allows to use it as the basis for development of training-methodological systems for various training courses possessing structure “tasks class – their solution methods” with the ability to exchange data between classes. Structure of a module responsible for a particular class of tasks is constructed in such way that the core can automatically detect not only the presence of the module, but also some of its characteristics (period of usage, number of attempts to use, etc.). The system is designed with possible enhancement of its functionality by adding new classes of tasks and methods for solving existing tasks. A new module project implementing abstract classes inserted at the design stage is creating to extend the list of classes of tasks to be solved. The formed project is compiled into a library that is added to the plug-in folder (padding) of the system. New function will be added automatically to the main menu of the projects after the software package start-up. Adding of new tasks solution methods is implemented slightly differently. It needs familiarization with the abstract class of a specific module. Each project of a module, as well as project pattern, has a special folder containing files with tasks solution methods classes for specific modules. Adding of a new method performs as addition of a new file in the folder where method class will be implemented, then the entire project must be recompiled. Since the system consists of a core and a number of modules provided in the form of plug-ins, the presence or absence of a single module does not affect the operation of other modules of the system. The scheme of the individual modules is shown in Figure 1.

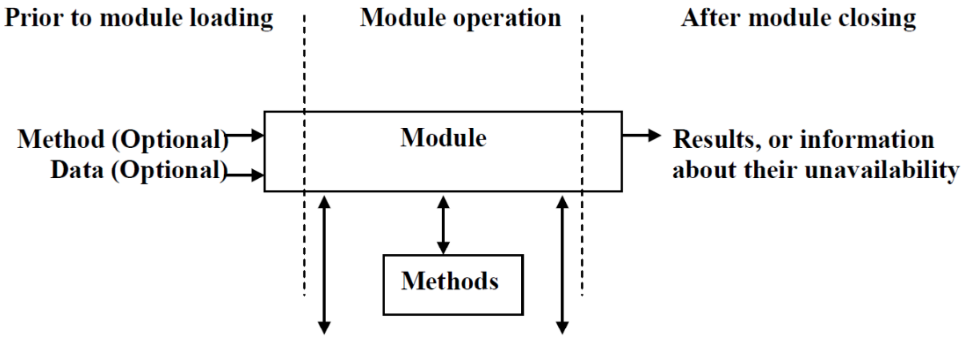


**Figure 1.** The scheme of the module

Each topic is presented as a separate module in a separate window. Data can be entered both from the keyboard and the program buffer. Each module has a test system designed to knowledge test. The system is laid down 8 DMT course sections (according to [Voloshyn, Maschenko, 2010]:

1. Conflicts and compromises; 2. Multi-criteria optimization; 3. Cooperative decision-making (DM); 4. Voting methods; 5. Expert evaluation, 6. DM in uncertainty conditions; 7. DM in fuzziness conditions; 8. Collective utility functions and topic "Psychological tests" [Karelin, 2007].

The authors hope that colleagues (teachers and students) from other universities (primarily in Ukraine and Russia) will use SMPR software system in the learning process, its toolkits (core) will be used to include new algorithms and improvements (redesign, treatment, substitution ) which are already available in the system.

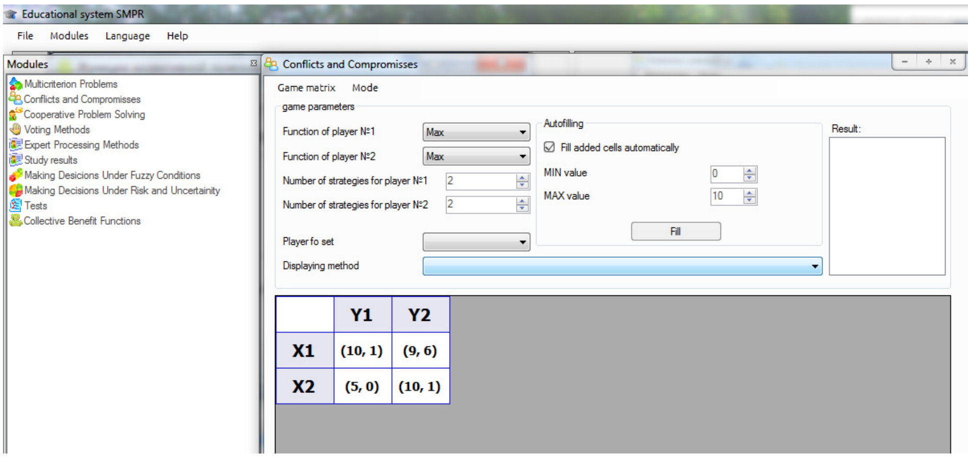
## The functionality of the main window of the system

"Crown" of the program includes bookmarks: 1) "File" allowing quick program closing; 2) "Modules" allowing to start any module; 2) "Language" allowing to switch interface to one of the four languages (Ukrainian, Russian, English, Chinese); 3) "Help", which has two divisions: 3.1) "About program" calling window with full information about developers of the project and their contacts; 3.2) "Modules" opening additional menu. It is attached to each of the connected modules, contains information about usage of the module and theoretical material.

## Example of operation of module “Conflicts and compromises”

Let’s consider operation and interface of the module "Conflicts and compromises" (Figure 2). Using options you can set the specification for task to be solved. Input matrix can be defined from keyboard, downloaded from clipboard, or automatically generated, if necessary. Module contains 10 methods each of them can be used. The solution is highlighted in green in the input matrix and displayed in a separate field for convenience.

Each module has a second mode - testing mode and there is an ability to choose methods for testing user's knowledge independently. To enter the answer "click" mouse pointer on the field containing the answer. Button "Check" will indicate correct result and displays the number of points earned. Then this result will be automatically stored in the core of the system for the demonstration of the level of knowledge, on user’s demand.


Educational system SMPR
File
Modules
Language
Help
Modules
Multicriterion Problems
Conflicts and Compromisses
Cooperative Problem Solving
Voting Methods
Expert Processing Methods
Study results
Making Desicions Under Fuzzy Conditions
Making Decisions Under Risk and Uncertainity
Tests
Collective Benefit Functions
Conflicts and Compromisses
Game matrix
Mode
game parameters
Autofilling
Fill
Result:

Function of player №1: Max

Function of player №2: Max

Number of strategies for player №1: 2

Number of strategies for player №2: 2

Player fo set:

Displaying method:

☑ Fill added cells automatically

MIN value: 0

MAX value: 10

|  | Y1 | Y2 |
|---|---|---|
| X1 | (10, 1) | (9, 6) |
| X2 | (5, 0) | (10, 1) |

**Figure 2.** Structure of the module "Conflicts and compromises"

All modules feature by random data filling for quick demonstration or knowledge testing. With user-friendly interface elements you can assign your own values and immediately get a solution. As it was mentioned, each module may take the value from the program buffer which stores an arbitrary number of values of scalar, vector, and matrix type.

In the tab "Help" there is all theoretical information (problem statement, solution methods, etc.) required for absorption of the material used in the module.

## Knowledge monitoring system

Each of the modules of software system is implementing methods offering student problem statement and giving a specific task (for example, finding the optimal strategies according to some criteria). Tasks are generated automatically (to avoid repetitions), and may have different complexity (problem dimension, number of states or participants, etc.).

All information about grades obtained during testing will be stored in the system and can be obtained by user at any time in a form of window with general testing results in all course sections and recommendations in topics to be studied better. All information about grades obtained during testing stored in the system and can be obtained by the user at any time. Window will appear with general results of the course and recommendations on the materials to be studied better.

As far as different modules comprising testing differ significantly from each other in the type and complexity of tasks, mathematical model taking into account parameters tasks is used when developing the system. Along with number of correct answers the following parameters as task complexity and performance are taken into consideration. Limiting values of allowed time and acceptable number of errors will be determined according to complexity of task. To account for the present time, the job evaluation in the form of *R(t)* = *k(t) R*, where *R* - evaluation without the runtime, *k (t)* - a factor which is defined as follows:

$$k(t)=\begin{cases} k_1, t<t_1 \\ -\frac{k_1}{t_2-t_1}\cdot t+(k_1+\frac{k_1}{t_2-t_1}\cdot t_1), t\in\left[t_1,t_2\right] \\ 0, t>t_2 \end{cases}$$

That is, if the task was completed faster than required time $t_1$, student will receives the highest rating, with time increase student will lose points till time $t_2$, after which he will receive no points for the task. The value $k_1$ was set to 1, but can be increased (student motivation to perform tasks faster to get more points). Values $t_1$ and $t_2$ are determined by experts (taking into account statistics on test results) for each module of the system taking into consideration complexity of the tasks.

The level of task complexity is set by the linguistic variable LV = {very low (VL), low (L), medium (M), high (H), very high (VH)}. Parameter "number of correct answers" is set by NCA variable = {very small (VS), small (S), medium (M), high (H), very high (VH)}. For example, the above values of linguistic variables can be set by intervals between 0% to 50% with discrete 10%. In intervals values of membership functions are linear with values from [0,1].

Let each "credibility value degree level" corresponds to a certain grade, for example, "4-point scale": "unsatisfactory", "satisfactory", "good", "excellent" (as a rule, such grades correspond to the following values in 100-point scale: [0.59] [60.74] [75.89] [90,100]).

SMPR system user is offered a "standard" base of rules for tested person evaluation output, containing, for example, rules of the type - "if VL = H and NCA = S, then grade = "unsatisfactory" with respective "credibility value degree" (determined by a given algorithm, fuzzy logic inference to be chosen by user from respective base [Zadeh, 1965; Snytyuk, 2008]).

The result of algorithm is a real number in the interval [0,1] or an integer number in the interval [0,100] - "the exact credibility value" for student knowledge grading.

Optionally, the user based on the received accurate ("clear") grade (CG) can get "fuzzy" grade (FG), having selected certain rules from a database of fuzzy inference rules (or through application of "standard" basis) of the following type: "if CG is in the range [60,65], your score = "solid three".

When using a "uniform" grading it can be possible that one student received 76 points out of 100 corresponding to the grading "good" and the other student - 74 points corresponding to the grading "satisfactory."

Despite the difference in only 2 points in the knowledge level, students get different grades, which may be a "cruel joke" in the problem in getting the scholarship. 5-point fuzzy evaluation is less sensitive to minor variations and more in line with teacher’s psychology (in the opinion of one of the authors of the article - former student). Membership function (MF) shall be used for fuzzy grading of the following form (Figure 3):

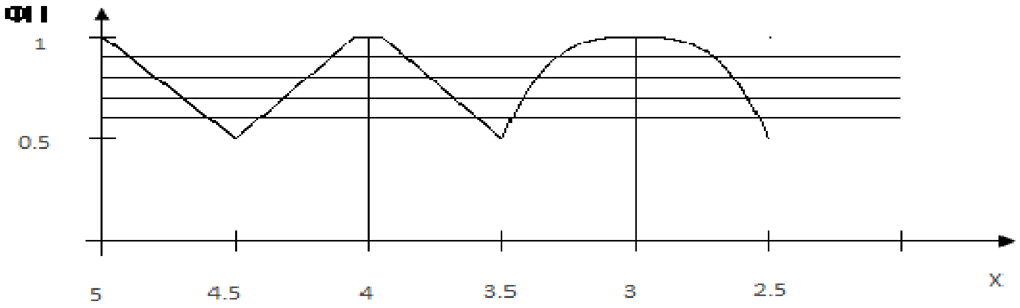


**Figure 3.** Membership function

MF form can be explained by the following reasons. When student receives "firm" excellent grade having solved all tasks or making one minor error, students can get "firm" good grade when have committed various errors. Fuzzy grade shell is defined as the closest to a clear grading with the level belonged to it. Depending on MF, let’s attribute the grading to one of 5 levels.

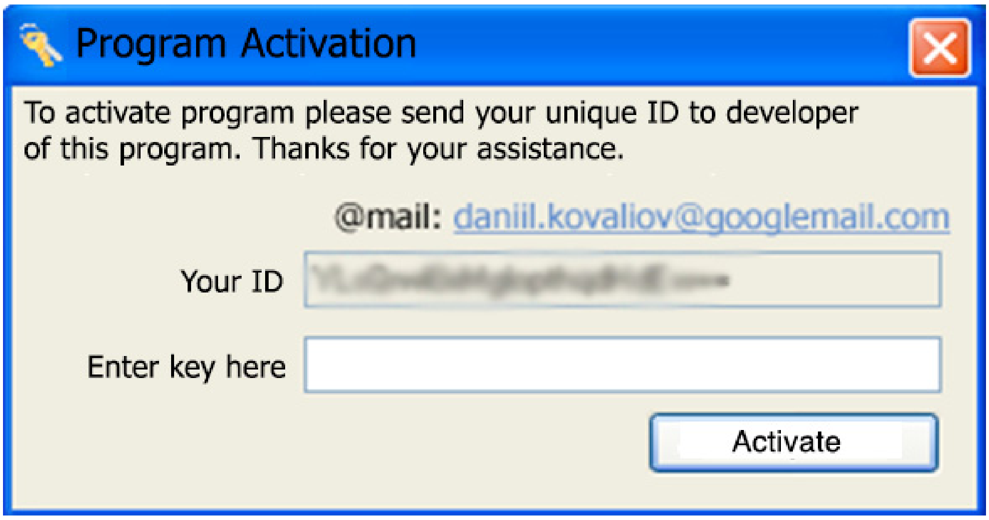
Thus, a high level (from 0.8 to 1) will correspond to "firm" grading, level (0.5, 0.7) – will correspond to grading "at a stretch." Correlation estimates with an "intermediate" level (0.7, 0.8) for a particular type depends on the psychosomatic characteristics of teacher [Karelin, 2007].

In determining the total grade in the training course we may form the rules basis of grading (or using the standard one built in SMPR system) with output like "you are almost good", for "good" grading you have to study better such-and-such topic".

## Protective system

Compared with the version SMPR-10, protection system limiting the possibility of unauthorized copying of software and allowing better care of users has been added to SMPR-12.

Features of the protection system consist of the fact that during the first start the program demands to send a unique identifier to the developer based on which key for the program (Figure 4) will be created. This key will be operating only with a computer that was obtained identifier.



Program Activation

To activate program please send your unique ID to developer of this program. Thanks for your assistance.

@mail: daniil.kovaliov@googlemail.com

Your ID [redacted]

Enter key here

**Figure 4.** Protection activation system

The protection system is implemented using readout of serial number of hard disk and its encryption in unique code, which the administrator receives from user for example by mail, and, based on it, creates activation key with a special program. Thus, the administrator has a database of the program users. It allows providing updates for them and rendering assistance at any time on application of the program.

## Development prospects

The concerned software is planned to be modified in such way that it will be available as an online resource for everyone. Thus, with only a link to the website, any student or teacher wishing to get acquainted with the course "The Theory of Decision Making", will be able to access the SMPR system - its theoretical, practical and test-pieces. Also, with the help of user registration system, it will be possible to perform evaluation of students remotely.

Video lectures recorded by a teacher will be added. They will explain in detail the theoretical part of the material with examples and useful information. Lectures will be subdivided into blocks of topics, each block of 4-5 videos with a total duration of about an hour: thus, mobility and accessibility of lectures be achieved to all comers.

Ideally, the program should be a part of an online educational project, which is being developed at the Kiev National Taras Shevchenko University, like the world-famous projects courser.org, udacity.com and others.

The program code and its platform for such modifications will be significantly changed, as well as web development technology must be enabled to provide GUI site for the functional part of the program.

## Acknowledgements

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA Bulgaria www.ithea.org, and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine www.aduis.com.ua.

## Bibliography

[Bondarenko, 2011] Бондаренко А. Применение обучающей программы SMPR в области сетей доступа. // Сборник научных трудов VII Всеукраинской научно-методической конференции «Современные технологии высшего образования». Одесса, 2011.–С.5 (in Ukrainian).

[Bondarenko, 2012] Бондаренко А. Применение обучающей программы SMPR при выборе технологий доступа // Problems of Computer Intellectualization.–Kyiv-Sofia: V.M.Glushkov Institute of Cybernetics, ITHEA, 2012.– P.332-339 (in Russian).

[Gaivoronskaya, 2013] G.Gaivoronskaya, A.Bondarenko, "The task of choosing the topological structure of the network access", Refrigeration and technology, № 2, 2013 (in Russian).

[Gamma, 2006] E.Gamma, R.Helm, J.Vlissides, "Elements of Reusable Object-Oriented Design. Design patterns", Moscow: Publishing House "Peter", 2006 (in Russian).

[Karelin, 2007] A. Karelin, "Great Encyclopedia of psychological tests", Moscow: Penguin Books, 416 p., 2007 (in Russian).

[Snytyuk, 2008] V. Snytyuk, "Forecasting. Models, methods and algorithms", Kiev, Maklaut, 364p., 2008 (in Ukrainian).

[Voloshyn, 2001] A. Voloshyn, S. Mashchenko, Методические рекомендации к выполнению практических и лабораторных работ по теории принятия решений. - Киев: Издательско-полиграфический центр «Киевский университет», 2001. - 46 с. (in Ukrainian).

[Voloshyn, 2006] A. Voloshyn, S. Mashchenko, "The theory of decision making. Textbook", Kiev: Publishing center "Kiev University", 304p. 2006 (in Ukrainian).

[Voloshyn, 2007] O.Voloshyn, K.Berezovskiy, I.Drozdov, "Developing collective teaching computer software for the course Decision theory", International Journal "Information Technologies & Knowledge, Vol.1, N1, pp.33-36, 2007.

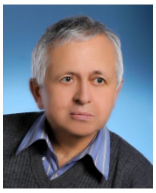
[Voloshyn, 2010] A. Voloshyn, S. Mashchenko, "Models and methods of decision-making. Textbook", Kiev: Publishing center "Kiev University", 336p., 2010 (in Ukrainian).

[Voloshyn, 2012] A. Voloshyn, D.Kovalev, "Educational and methodical testiruyusche-evaluator software system to support courses in decision theory", Problems of Computer Intellectualization, ITHEA, pp.293-298, 2012 (in Russian).

[Zadeh, 1965] Zadeh L., "Fuzzy sets", Information and Control, 1965. - P.338-353.

[Соломицкий, 2012] Соломицкий М., Болотина О. Опыт использования методов решения задач выбора в условиях неопределенности и риска при проектировании компьютерных сетей // Problems of Computer Intellectualization. – Kyiv-Sofia: V.M.Glushkov Institute of Cybernetics, ITHEA, 2012.– P. 318-331 (in Russian).

## Authors' Information



***Oleksii Voloshyn** – Professor, Taras Shevchenko National University of Kyiv, faculty of cybernetics. Kyiv, Ukraine, 01017 Vladimirskaja str. 64; e-mail: olvoloshyn@ukr.net*

*Major fields of scientific research: decision making, decision support systems, mathematical economics, expert systems, fuzzy analysis , e-learning*

***Daniil Kovaliov** – post-graduced student, Taras Shevchenko National University, faculty of cybernetics.; e-mail: daniil.kovaliov @ gmail.com*

*Major fields of scientific research: decision making, decision support systems, expert systems, e-learning*