AUTOMATING THE EVALUATION OF EDUCATIONAL SOFTWARE

Ioannis Stamelos[1], Ioannis Refanidis[1], Panagiotis Katsaros[1], Alexandros Tsoukias[2]
Ioannis Vlahavas[1] and Andreas Pombortsis[1]
[1]Dept. of Informatics, Aristotle University, Thessaloniki, 54006, GREECE
{stamelos, yrefanid, katsaros, vlahavas, pombortsis}@csd.auth.gr
[2]LAMSADE-CNRS, Universite Paris-IX, Dauphine, Paris Cedex 16, France
tsoukias@lamsade.dauphine.gr

ABSTRACT

This paper proposes a framework for educational software evaluation based on the Multiple Criteria Decision Aid methodology, supported by ESSE, an Expert System for Software Evaluation. An evaluation example is presented that illustrates the overall evaluation process. Evaluating educational software products is a twofold process: both the technical and the educational aspect of the evaluated products have to be considered. As far as the product's educational effectiveness is concerned, the flexibility of ESSE in problem modeling allows the development and the use of a set of criteria, which clearly describe the context, and the educational setting in which the software products are to be used. From the technical point of view, a software attribute set based on the ISO/IEC 9126 standard has been chosen together with the accompanying measurement guidelines.

INTRODUCTION

Evaluating software products is a particularly difficult process because many, often contradictory, criteria have to be taken into account. An important effort for defining a universally accepted model has been done by the International Standard Organisation (ISO), which has published the ISO/IEC 9126-1, 9126-2 and 9126-3. ISO proposes six attributes, which assesses the quality of a software product: *functionality*, *reliability*, *usability*, *efficiency*, *maintainability* and *portability* [1]. These attributes can be further analyzed in lower-level attributes.

However, ISO does not cope with the definition of software attributes appropriate for assessing product quality from a non-technical point of view. In the case of educational software it is generally accepted that it is very difficult to develop a predefined set of standards according to which the educational value of the software can be defined. The reason is that each educational software product does not necessarily serve the same learning objectives and the same target users (age, level of knowledge or skills). For this reason the set of criteria to be chosen for assessing the educational value of a software product must clearly prescribe the evaluation context in each case.

This paper presents an evaluation framework for educational software products based on the Multicriteria Decision Aid methodology (MCDA) [5, 7], which is suitable for evaluation problems where many criteria have to be taken into account. The ISO/IEC 9126 was chosen as the basis for evaluating the quality of a software product from the technical point of view while an adaptable set of criteria is proposed for assessing the educational value of the product.

The proposed evaluation framework has been processed with ESSE, an expert system for software evaluation that supports various MCDA methods [8]. The main capabilities of ESSE are the following:

- Partial automation of the software evaluation process.
- Suggestion of a software evaluation model, according to the type of the problem.
- Support of the selection of the appropriate MCDA method, depending on the available information.
- Assistance provided by expert modules, called throughout this paper *Expert Assistants*, which help the evaluator in assigning values to the attributes of the software evaluation model.
- Consistency check of the evaluation model and detection of possible critical points.
- Management of past evaluation results, in order to reuse them in new evaluation problems.

In the following section we discus the educational software evaluation process. Next we give an example of an evaluation session using ESSE and we present the advantages of the inclusion of a knowledge-based system in the evaluation process. The appendix presents in detail the set of attributes chosen for the educational part of the evaluation.

EDUCATIONAL SOFTWARE EVALUATION

In order to evaluate an educational software product a set of attributes is needed. These attributes are organized in a tree-hierarchy, where the higher level attributes describe general aspects of the evaluated products, while the lower level attributes deal with more specific aspects of the evaluation. Each one of the higher level attributes is decomposed in a number of sub-attributes. The lower level attributes, that are not further decomposed, are called ‘basic attributes’ while the higher level attributes are called ‘compound attributes’. Each basic attribute is assigned a scale and a measurement method. The scale of the method can be arithmetic or nominal while in the latter case an ordering between the possible values has to be defined

As already mentioned, evaluating educational software is a twofold process, since both the technical and the educational aspect of the evaluated products must be considered. Therefore, the proposed framework consists of two top-level attributes, one concerning the technical features of the evaluated products and one concerning the educational effectiveness of them. In the next paragraphs we present these two major attribute sub-trees and briefly the main steps of MCDA methodology.

Attributes for Evaluating the Technical Features

ISO 9126 is used as a basis for assessing the quality of educational software products from the technical point of view. Quality is decomposed in six sub-attributes, and each one of them is further decomposed in sub-sub attributes in the following way:

- ‘Functionality’ [‘suitability’, ‘accuracy’, ‘interoperability’, ‘compliance’, ‘security’]
- ‘Reliability’ [‘maturity’, ‘fault tolerance’, ‘recoverability’, ‘availability’]
- ‘Usability’ [‘selectability’,‘learnability’,‘operability’]
- ‘Efficiency’ [‘time behavior’, ‘resource utilization’]
- ‘Maintainability’ [‘analyzability’, ‘changeability’, ‘stability’, ‘testability’]
- ‘Portability’ [‘adaptability’, ‘installability’, ‘conformance’, ‘replaceability’]

ISO 9126 standard offers an initial decision model, which may be adapted to the characteristics of a specific evaluation problem. However, the applicability and the significance of each one of the ISO 9126 specified attributes in a software evaluation process depend strongly on the context and the type of the evaluation problem.

Attributes for Evaluating the Educational Effectiveness

In contrast with the technical aspect of the evaluation, there is no broadly accepted model for the educational aspect of the evaluation. The reasons for this are mainly:

- It is very hard to describe the context of all possible educational software evaluation problems with a single attribute framework. For example, the evaluation carried out by a teacher or a trainer is a completely different problem compared to the evaluation process carried out by a decision-maker of an educational institution. In addition, factors that must be taken into account are the type of target users the evaluator has in mind while undertaking the evaluation and the way he or she intends to use the software (for example, to teach a specific topic, or to enhance students’ understanding of a certain topic).

- There are several types of educational software products. According to [2] these types are: ‘drill and practice’, ‘tutorials’, ‘simulations’, ‘instructional games’ and ‘problem solving’. Each of these types may need different evaluation criteria.

- An educational software product may have such original characteristics that prevent the use of a predefined set of evaluation criteria.

For the purpose of our study we have tried to take into consideration all elements relevant to teachers, trainers, parents and users. However, the proposed set of criteria must be viewed as a general evaluation framework that will most certainly need modification.

The framework we propose is based on the work presented in [6], which we have modified by removing the attributes related to the technical aspect of the evaluation (since for the technical aspect we use the ISO standard) and extending in more detail the attributes related to the educational aspect of the evaluation. According to our framework the educational effectiveness attribute of a software product is decomposed in two sub-attributes, where each one of them is further decomposed in sub-sub-attributes. The first two levels of this decomposition are shown in the table 1.

• ‘educational features’
- ‘target users specification’ - ‘information for the topics addressed and the learning objectives’ - ‘instructional support materials’ - ‘adaptation to individual needs’ - ‘strategies for enhancing engagement, attention and memory’ - ‘usage of the product’ - ‘encouragement of critical thinking’ - ‘user performance assessment’
• ‘content’
- ‘quality of content’ - ‘appropriateness’ - ‘structure’

Table 1: Educational effectiveness decomposition

Appendix A presents in more detail the attributes proposed for the educational aspect of the evaluation, together with a brief description of each one of them.

Multiple Criteria Decision Aid methodology (MCDA)

This paragraph describes briefly the factors involved in an MCDA evaluation and the procedure followed for MCDA application. An evaluation problem solved by MCDA can be modeled [7] as a n-ple {*A,D,M,E,G,R*} where:

– *A* is the set of alternatives under evaluation in the model
– *D* is the set of the evaluation attributes
– *M* is the set of associated measures
– *E* is the set of scales associated to the attributes
– *G* is the set of criteria constructed in order to represent the user's preferences
– *R* is the preference aggregation procedure

In order to solve an evaluation problem, a specific procedure must be followed. This procedure consists of seven steps [3, 7]:

– Step 1: Definition of the evaluation set *A*
– Step 2: Definition of the type of the evaluation
– Step 3: Definition of the set of evaluation attributes *D*
– Step 4: Definition of the set of measurement methods *M*
– Step 5: Definition of the set of measurement scales *E*
– Step 6: Definition of the set of Preference Structure Rules *G*
– Step 7: Selection of the appropriate aggregation method

Notice that the order of execution of the above steps is not strict. For example, it is possible to define first the set *D* and then, or in parallel, define *A*, or even select *R* in the middle of the process. More details about the application of MCDA in evaluation problems are given in [7].

EVALUATION PROCESS WITH THE ASSISTANCE OF ESSE

This chapter describes a hypothetical evaluation of N educational software products. The purpose is to illustrate how ESSE [8] is involved in the evaluation process and therefore the example is not bound to a specific evaluation problem. Additionally, we limit the criteria to those most closely related to the educational software (e.g. no 'cost' criteria will be considered). In order to solve the evaluation problem, we follow the seven-step MCDA procedure described earlier. At each step we describe how ESSE is involved in the evaluation process.

The first step of the evaluation process is the definition of the evaluation set *A*. ESSE represents the products under evaluation in its knowledge base as instances of a generic frame named 'product'.

The second step is the definition of the type of the evaluation. This type characterizes the form of the desired outcome. The most known types are *classification*, *choice*, *sorting* and *description*. Currently ESSE supports only classification. This type is the most general, since the elements of *A* are ranked from the best to the worst.

The third step is to define the set of attributes *D*. The knowledge base of ESSE contains frameworks for several categories of software evaluation problems. By selecting the category 'educational software', the attribute framework presented in the previous section is retrieved. In the case that the knowledge base contains already solved evaluation problems of the same type, the system presents their characteristics (attribute hierarchy, weights, scales etc.) and prompts the user to either select one of them that suits better his current problem or proceed with the definition of a new solution. The user can modify the proposed attributes, removing some of them or adding new ones and he may accept or modify the weights proposed for the various attributes and the scales for the basic ones.

To each one of the basic attributes a measurement method must be assigned (step 4). ESSE's expert assistants propose some measurement methods. The user can accept the guidance of ESSE or define his own measurement methods. The measurements obtained according to a measurement method have to be transformed in appropriate scale values (step 5). There are two types of scales, the *arithmetic* and the *ordered nominal* scales. If an outranking aggregation method (for example ELECTRE II, [4]) is to be used, preference structure rules have to be defined. These rules will determine the superiority of a product against another, with respect to a specific attribute (step 6). For example, such a rule could determine that a product is better than another with respect to the attribute 'quality', if it is superior in the 70% of the quality's sub-attributes (taking into account their relative weights). ESSE gives the user the ability to define such rules for each one of the attributes.
After constructing the evaluation model, the system suggests the use of ELECTRE II (step *7*). This is the result of the activation of two rules from the MCDA method selection knowledge base. The first rule detects that the top-level attributes are only two ('quality' and 'educational effectiveness') and suggests the use of a method which employs weights. The second rule detects that the model has nominal basic attributes and suggests the use of an outranking method. The method that fulfills these requirements is the ELECTRE II, which is both an outranking method and supports weights.

Finally, values are assigned to the basic attributes for each product, using the measurement methods selected in step 4 and the aggregation method selected in step 7 is carried out, obtaining the desired results. Outranking methods, like ELECTRE II, rank the evaluated products in an order, without giving any information about their absolute distance. The final result may be of the form:

Product1 > Product2 = Product3 > Product4 ...

After completing the above example, the entire problem and its solution is saved in the knowledge base of the expert system. This knowledge will be used by ESSE in future evaluation problems of the same type. It is obvious that the more the knowledge base of ESSE is enhanced with new instances of educational software evaluation problems, the greater the assistance ESSE will offer to future evaluators.

CONCLUSIONS AND FUTURE WORK

In this paper a framework for educational software evaluation is proposed, which takes into account both the technical and the educational aspect of this type of software products. For the technical part of the evaluation the ISO 9126 standard is adopted. For the educational part of the evaluation, it seems that it is not possible to define a single set of attributes appropriate for any problem. The attribute framework to be used depends on the type of target users the evaluator has in mind, on the way he or she intends to use the software and on the instructional strategy that has been chosen. Although a quite general set of attributes based on the ideas of [6] has been proposed, it seems to be more important to support the adaptation of the proposed set of attributes or to support the development of an entirely new attribute framework by preserving the ability of reusing existing problem solutions.

The evaluation is performed with the Multiple Criteria Decision Aid methodology, which is suitable for evaluation problems where many criteria have to be taken into account. The evaluation process is supported with ESSE, an Expert System for Software Evaluation. ESSE assists the evaluation process by suggesting an evaluation framework, according to the type of the problem. Moreover, it supports the selection of the appropriate MCDA method and it manages and re-proposes past evaluation problem instances, in order to be reused in new evaluation situations.

In the future we will continue working on the proposed framework, applying it in a sufficiently large number of cases. Moreover, we plan to explore the applicability of more MCDA methods, such as other outranking and multiple attribute utility methods, some interactive methods, etc. In addition, we will explore the applicability of these methods to other categories of software evaluation problems, obtaining additional rules of experience. Finally, it is planned to maintain the knowledge bases, by inserting new findings in software engineering practice and by applying ESSE to numerous software evaluation problems of different types.

REFERENCES

[1] ISO/IEC 9126-1 (1996), Information Technology - Software quality characteristics and sub-characteristics.

[2] Lockard J., Abrams P. and Many W. (1987), *Microcomputers for educators*, Little, Brown and Co., Boston.

[3] Morisio M. and Tsoukias A. (1997), IusWare, A methodology for the evaluation and selection of software products, *IEEE Proceedings on Software Engineering,* 144, 162-174.

[4] Roy B. and Bertier P. (1973), La methode ELECTRE II - Une application au media planning, *in OR72*, M. Ross (ed.), North Holland, Amsterdam, 291-302.

[5] Roy B. (1996), *Multicriteria Methodology for Decision Aiding*, Kluwer Academic, Dordrecht.

[6] Antonio Ulloa Severino (1998), Educational Effectiveness Evaluation Criteria, *Emerging New Technologies in Education*, Samos, Greece.

[7] Vincke P. (1992), *Multicriteria decision aid*, Wiley, New York.

[8] Vlahavas I., Stamelos I., Refanidis I., Tsoukias A. (1998), *ESSE: An Expert System for Software Evaluation*, to be published

APPENDIX A

Educational Features

➢ Target users specification: The software packaging or the accompanying reference materials must clearly inform about the approximate age of the target users and about the prerequisite level of knowledge or skills recommended for best use of the software.
SCALE: fully specified > partially specified > not specified.

➢ Information for the topics addressed and the learning objectives: It is very important that instructors and educators are provided with clear and comprehensive information concerning both the topics that the educational software deals with and the learning objectives that it aims to achieve. Obviously, the topics addressed by the software must be relevant to the set learning objectives, so as to enable users to achieve them, and the learning objectives must be appropriate for the target users' age and competence. When the educational software is designed for classroom use to ensure that the software is a valuable educational resource, the topics covered and the learning objectives must be compatible with the education system of the country where it is used.
SCALE: fully specified & consistent > fully specified but not consistent > partially specified > not specified.

➢ Instructional support material: Another aspect to take into account when evaluating the educational features of a particular piece of software is the quality of the instructional support material it provides, either in print and/or as printable files from disc or on-line resources. In

fact, they can significantly help not only instructors but also users to focus the potentialities of the software, giving suggestions on the various teaching strategies instructors can adopt using it in the classroom, informing about how the program can be fitted into a larger framework of instruction etc.
SCALE: adequate & complete > not complete > not appropriate or not clear enough > not existent.

➢ Adaptation to individual needs:
Feedback: The software product provides feedback not stereotyped, but appropriate for the situation and the users' performance.
SCALE: feedback appropriate for each different situation > stereotyped feedback > no feedback.
Possibility to follow different learning routes (exploratory learning environments): The software product is important to allow the users to follow different learning routes through the program.
SCALE: possible > not possible
Differentiate the level of difficulty in respect with the user's performance:
SCALE: possible > not possible
Level of interactivity:
SCALE: good > not so good > bad
➢ Strategies for enhancing engagement, attention and memory:
User motivation: User motivation can be enhanced in the following ways:
- Show to the users the usefulness of what they learn.
- Set clear goals (e.g. number of questions that need to be completed without a mistake) and provide indication of how the user is proceeding periodically.
- Encourage users to envision themselves in an imaginary context or event where they can use the information they are learning.
- Cognitive curiosity: giving partial information, elements of surprise, stimulating desire to know e.t.c.
- Sensory curiosity: sound, visual stimuli e.t.c.
- Provide a level of user control, keeping always in mind that too much user control can be detrimental.
- Confidence: provide reasonable opportunity to be successful.
- Competition with other users (students)
- Competition with the computer
- Competition with the user him/herself
- Competition with the clock
- Adjunct reinforcement: Follow the successful completion of any activity with an activity that the user (student) finds enjoyable.
SCALE: good > not so good > bad
Varied tasks & activities:
SCALE: varied tasks & activities > monotonous routines
Retention of information: Retention of information is encouraged when the difficulties are well distributed throughout the program, the topics are clearly connected and summaries of the main topics covered in each preceding section are provided.
SCALE: good > not so good > bad

➢ Usage of educational software: It is very important to consider the possible usage of the educational software as learning resource in the classroom or by a single user as self-instructional resource, if it can be useful for the administration of tests, or it can be used only for instructor-led tuition.
SCALE: many cases of usage > only one possible usage

➢ Encouragement of critical thinking: It must be taken into account if the program provides critical thinking and decision making activities that entail inductive or deductive reasoning and problem-solving skills.
SCALE: existent > not existent

➢ User's performance assessment: For true and actual learning to take place, it is important that the educational software allows the users to constantly monitor and assess their learning progress.
SCALE: different types of assessment activities > only one type of assessment activities > no assessment activities

Content

➢ Quality of content
Accuracy:
SCALE: accurate > contains inaccuracies.
Clear formulation of the content so as to be easily understandable:
SCALE: clear formulation of the content > not so clear formulation of the content
Complete: Whether or not the software is complete in dealing with all the aspects of each topic?
SCALE: complete > incomplete
Up-to-date:
SCALE: up-to-date > relatively old > old.

➢ Appropriateness: This attribute refers to the appropriateness of the reading level for the target users. Users should be able to understand the information presented, so it is essential to check if vocabulary, structure and sentence length are suitable for their level of knowledge, presenting an acceptable degree of difficulty.
SCALE: appropriate > not appropriate

➢ Structure: This criterion focuses on the organization of content, which should be logically structured and divided among the sections or modules, in order to help the user to progressively assimilate information.
SCALE: modular structure > linear structure > unstructured.