A FRAMEWORK FOR EDUCATIONAL SOFTWARE DEVELOMENT EVALUATION

Paula Escudeiro

Departamento de Engenharia Informática, Instituto Superior de Engenharia do Porto, Rua Dr António Bernardino de Almeida, 431, 4200-072 Porto, Portugal

José Bidarra

Departamento de Ciências Exactas e Tecnologia, Universidade Aberta, Rua Escola Politécnica, 147, 1269-147 Lisboa, Portugal

Abstract: *In this paper we propose a Quantitative Evaluation Framework used to evaluate educational software systems built with X-TEC (Techno-Didactical Extension for Instruction/Learning Based on Computer) model, in order to validate and strengthen the potential quality of e-Learning systems.*
The Quantitative Evaluation Framework and the Techno-Didactical Extension for Instruction/Learning Based on Computer model are based on the paradigms of software engineering applied to the construction of educational software.

Introduction

Modern information and communication technology renders new ways of organizing teaching and learning processes, which are independent of time and space.

E-learning has been described as the ´use of electronic technology to deliver, support and enhance teaching and learning` [Learning Technologies, 2003].

Despite the theoretical benefits that e-learning systems can offer, difficulties can often occur when systems are designed with out consideration of learner's characteristics [Frredman and Liu, 1996; Liang and McQueen, 1999].

In general, educational software systems are based on methodological approaches which are fundamentally concerned with processes or data. Their lifecycle is supported on two different and independent stages: instructional design and technical development.

The gap between the typical skills and terminologies of these two stages usually leads to a problem: the final product is far away from the initial requirements proposed by the author. Consequently, these approaches usually imply the high risk of obtaining low quality products. The X-TEC model [Paula Escudeiro, José Bidarra, 2006] tries to solve this problem.

To evaluate the educational software systems based on X_TEC model we propose a generic Quantitative Evaluation Framework (QEF). This framework may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between different tools.

Quantitative Evaluation Framework

X-Tec model is supported by software engineering goals, principles and actions [Pressman, 2001], [Bates, 2000].

To evaluate the educational software systems based on X_TEC model we propose a generic Quantitative Evaluation Framework (QEF) [Paula Escudeiro, José Bidarra, 2006]. This framework may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between different tools.

Educational software quality (ISO 9126 is the standard of reference) [Scalet et al, 2000] is evaluated on a three dimensional space.

A dimension aggregates a set of factors. A factor is a component that represents the system performance from a particular point of view.

The dimensions of our Cartesian quality space are: Functionality (F); Efficiency (E) and Adaptability (A), represented in fig 1.

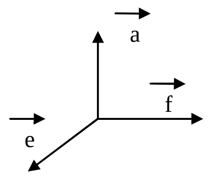


Fig 1: Cartesian quality space

For the evaluation of educational software systems based on the X_TEC model we propose a generic quantitative evaluation framework. This framework may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between distinct tools.

The quality q, of a given system is defined in our tri-dimensional Cartesian quality space, Q, and measured, in percentage, relatively to a hypothetically ideal system, I, represented

in our quality space by the coordinates (1, 1, 1).

Quality dimensions

The quality space, Q, aggregates, in the dimensions – Functionality; Efficiency and Adaptability – a set of factors that measure the relevant characteristics of an ESDM.

The Functionality dimension reflects the characteristics of the educational software related to its operational aspects. It aggregates four factors: feasibility, inviolability, easy of use and integrity

The Efficiency dimension aggregates four factors: data structure, programming structure, learning objects, imperfections recovery.

Through this dimension we measure the system’s ability for presenting different views on the course content with minimum effort.

The Adaptability dimension is the aggregation of five factors: flexibility modularity, reusability, scalability and maintainability. Through them we can measure to what extend the scenario and course content are efficacious – whether they are focused and able to present different instructional design theories and different learning environment in a common platform.

The quality for a given system coordinates may be obtained through the application of one of several aggregation forms. We will compute these coordinates as the average of the factors that contribute to it; the average is simple and gives the same relevance to all factors. Quality dimensions are based on the following factors:

Functionality	**Efficiency**	**Adaptability**
• Feasibility	• Data Structure	• Flexibility
• Easy of use	• Programming Structure	• Modularity
• Integrity	• Management Contents	• Reusability
• Inviolability	• Interface	• Scalability
		• Maintainability
		• Portability

For each system being developed we will have to identify the importance of each factor to the dimension. The dimension coordinate is them computed as the weighted mean of these factors:

Dimension i = $\sum_{n}(p_n \times factor_n)$, $\sum_{n}(p_n) = 1$ and $p_n \in [0,1]$

Where:
n is the number of relevant factors for the dimension.
Each factor is evaluated by:

Factor n = $\frac{1}{\sum_{m} pr_m} \times \sum_{m}(pr_m \times pc_m)$

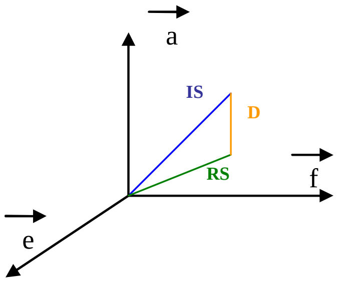
Where:
M is the number of valid requirements for the factor.
pr m is the weight of the requirement m
pc m is the fulfillment percentage of the requirement m.
The dissimilarity between the system under evaluation and ideal system is measured by:

D = $\sqrt{\sum_{j}\left(1 - \frac{Dim_j}{100}\right)^2}$

Finally the quality of the system is computed as:

Q = $1 - \frac{D}{\sqrt{n}}$, Q $\in [0,1]$
or
q = $\left(1 - \frac{D}{\sqrt{n}}\right) * 100$, q $\in [0,100]$

The quality of a system is measured as the distance between the ideal system (projected system) and the real system (final system).



The system quality is in the inverse proportion of the distance between the Ideal System (IS) and the Real System (RS).

If D=0 Then Q=1

If D=maxim, D $_{max} = \sqrt{n}$
Then Q=0

The measure of the system quality is obtained from a six steps process:
1st – Requirement classification
2nd – Factor classification
3rd – Result evaluation
4th – Dimension performance
5th – Global deviation
6th – System quality

Requirement Classification

The ideal system has a set of requirements that indicates what the system must do, see fig 2.
We start by associating weights to requirements, [0,1] based on the relevance of the requirement for that particular dimension, according to:

10 – Fundamental
8 – Very Important
6 – Important
4 – Necessary
2 – Optional
0 – Irrelevant

Requirements

Dimension

Functionality	Efficiency	Adaptability
Pr n	Pr n	Pr n
Pr n	Pr n	Pr n
SUM()	SUM()	SUM()

Fig2: Matrix of the dimension requirements

Factor Classification

Each factor contributes to the dimension value. This contribution is represented by a real number, Pn , between 0 and 1, indicating the relevance of the factor to the dimension. The dimension value is a weighted mean the factor that contributes to that dimension

Dimension = $\sum_n (p_n \times factor)$, $\sum_n (p_n) = 1$ and $p_n \in [0,1]$

Result Evaluation

It is very important to validate the requirements, so that system performance can be accurately evaluated.
The matrix in fig 3 shall be fulfilled during the evaluation process. Once it is completed the system quality is automatically computed.

Requirements

Dimension

Functionality	Efficiency	Adaptability
Pcm	Pcm	Pcm

Pcm	Pcm	Pcm

Fig 3: matrix of the factors

Dimension performance

The performance of a dimension is obtained through, the factors of each dimension.

Factor n = $\frac{1}{\sum_m pr_m} \times \sum_m \left(pr_m \times pc_m \right)$

And the dimension performance is given by:

Dimension = $\sum_n \left(p_n \times factor \right)$, $\sum_n \left(p_n \right) = 1$ and $p_n \in [0,1]$

Global deviation

The global deviation is obtained as the Euclidean distance between our system coordinates and the ideal system, whose coordinates are (1,1,1)

D = $\sqrt{\sum_j \left(1 - \frac{Dimj}{100} \right)^2}$ Global deviation

System Quality

The system quality is computed by:

Q = 1 - $\frac{D}{\sqrt{n}}$, Q $\in [0,1]$

q = $\left(1 - \frac{D}{\sqrt{n}} \right)$ * 100 q $\in [0,100]$ System Quality

We say that system quality is q% which means that the system is able to perform q% of its initial specifications.

Conclusions

In this work we propose a method to measure quantitatively the quality of a given educational system.

Quality evaluation frameworks, like the one we propose here, are crucial to help validating educational systems and ensure that they are adequate and follow the original specifications, before using them in the learning environment.

We are already applying X_TEC, for the development of educational software systems with our students, and using the quality evaluation framework to evaluate them. Our purpose is to realize the ability and applicability of our quantitative evaluation framework in real world solutions.

The QEF may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between different tools.

References

A. Dias Figueiredo (2005)	Context Engineering: And is Development Research Agenda, Universidade Coimbra, 2005
Allesi, S. e Trollip (1985)	S. Computer Based Instruction: Methods and Development, Prentice Hall, Inc. Englewood Cliffs, New Jersey, USA, 1985
Bates Tony, (2000)	A. W. Tony Managing Technological Change: Strategies for College and University Leaders. San Francisco, 2000
[4] BloomBertram B. Mesia, and David R. Krathwohl, (1964)	Bloom B. S. Taxonomy of Educational Objectives,: The Affective Domain & The Cognitive Domain. New York: David McKay Co Inc.
[5] Booch, (1994).	G. Booch, Object Oriented Analysis and Design With Applications, Second Edition, Benjamin/Cummings, Menlo Park, Califórnia, 1994.
[6] Clark 1994, Gery, (1994)	Gery, GJ Making CBT happen. Boston: Weingarten. Clark, RE Media will never influence learning. Educational Technology Research and Development
[7] Coad and Yordon, (1991)	P. Coad and E. Yourdon, OOA –Object Oriented Analysis, 2nd Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1991
[8] Crossley, K. e Green (1990)	Le Design des Didacticiels: Guide Pratique pour la Conception de Scénarios Pédagigiques Interactifs. ACL-Editions, Paris France 1990
[9] Eckerson (1995)	Eckerson, Wayne W. "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." *Open Information Systems 10,* 1 (January 1995): 3(20).
[10] Escudeiro, Paula; Bidarra José, (2006)	X-TEC: Techno Didactical Extension for Instruction/Learning Based on Computer, Orlando, Florida, SITE 2006.
[11] Gagné, (1996)	Gagné, Robert M. and Medsker, Karen L. (1996). *The Conditions of Learning Training Applications.* Florida: Harcourt Brace & Company.
[12] Jacobson, (1992)	Jacobson, Object Oriented Systems Engineering, Addison-Wesley, 1992.
[13] Keller/Back (2004)	Keller, M.; Back, A.: Blended-Learning-Projekte im Unternehmen, Learning Center der Universität St. Gallen, St. Gallen, 2004
[14] Merrill, (1981)	Merrill, M.D., Kowallis, T., & Wilson, B.G. Instructional design in transition. In F. Farley & N. Gordon (Eds.), Psychology and education: The state of union
[15] Minken, I., Stenseth, B. E Vavik L., (1998)	Educational Software. ULTIMA-Gruppen A/S, Haden, Norway, 1998
[16] Pressman Roger S., (2001)	Pressman, Roger S. Software Engineering a Practitioner’s Approach, 5 th Edition, McGraw-Hill Companies Inc, 2001
[17] Purinima Valiathan, (2005)	ASTD-Linking People, Learning & Performance. Learning circuits- American Society for Training & Development
[18] Rumbaugh et al, (1991)	J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
[19] Scalet et al, 2000	ISO/IEC 9126 and 14598 integration aspects: The Second World Congress on Software Quality, Yokohama, Japan, 2000.
[20] Yourdon, (1998)	E. Yourdon, *Managing the System Life Cycle*, 2nd Edition, Yourdon Press/prentice Hall, Englewood Cliffs, New Jersey, 1998.