Towards a formalization of the automatic generation of exercises

Sergio Gutiérrez[2], Francisco J. Losa[1]
and Carlos Delgado Kloos[1]

[1] University Carlos III of Madrid
Avda. de la Universidad 30
E-28.911 Leganés, Madrid (SPAIN)
[2] London Knowledge Lab
Birkbeck College
23–29 Emerald St.
WC1N 3QS London (UK)

Abstract. An adaptable educative system needs a flexible assessment strategy in order to be able to adapt to many different kinds of students. However, creating a big amount of different exercises with different presentations, levels of difficulty, etc, is usually a very time consuming task. Many attempts have existed for creating big numbers of exercises automatically, but they tend to be strongly linked to some domain. This paper tries to extract some common foundation from them to build a common theory, getting to a simple formalization of the concept of parametric exercise. The main idea is to have a core exercise that is different every time it is displayed to the student, being this change controlled according to some parameters. The paper describes the four different types of parameters that exist in such an exercise, and presents an authoring tool and implementation that has been created according to these ideas.

1 Introduction

In every adaptive e-learning system, the process of assessment is extremely important. Assessment is needed to acknowledge the progress of the learners, both from the system and from the own learners' point of view: checking her skills against a set of exercises allows the user to become aware about her understanding of the concepts explained by the system; on the other hand, many processes on the system side depend up to some degree on the assessment process, like user modelling [1] or sequencing [2].

There is a clear need for big sets of exercises in educational hypermedia systems. First, a limited or fixed set of tasks could be gamed by the

(i.e. external parameters) are: the disk size (in number of cylinders), the number of requests, the value of the requests, the initial position of the disk head and the disk access scheduling algorithm. Additionally, it is possible to specify the minimum and maximum number of cylinders in the disk, the exact disk size, the initial position of the head unit, the minimum and maximum number of requests, a specific number of requests, and the disk access scheduling algorithm. The difficulty of this problem can be set by modifying the values of parameters such as the number of requests or the scheduling algorithm, as the complexity of the solution grows linearly with the number of requests.

5 Conclusions and futute work

A parametric exercise is an exercise that can be replicated an unlimited number of times with different data. This data makes the appearance, difficulty and solutions different on each instantiation, adapting them to the particular needs of every student. Parametric exercises allow for the creation of a high number of instances without any repetition, being very useful for training applications or for any web-based system that requires a lot of input about the knowledge of the student.

This paper has presented a first step towards the formalization of this concept, describing the four types of parameters that are relevant: external, internal, answer and output. This is a small step in the process of creating a theory on automatic creation of exercises. Such a theory could open the way to a standardization effort. Standards are important because they promote reuse of learning material. In the case of static exercises, the IMS QTI specification has achieved a great level of success. This paper has presented an implementation and authoring tool that are based on these ideas. The paper does not address other important processes like sequencing or user modelling.

Next steps in this research move along two main lines. The first one aims at making it easy for designers without a technical background to create complex exercises. Our current implementation requires programming skills on the part of the exercise designer, being its main limitation. A possible solution may involve the use of script or iconic programming languages. The second line of research studies how QTI and the PE paradigm can be merged, in order to provide additional adaptation capabilities to QTI.

References

1. Wessner, M., Pfister, H.R.: Group formation in computer-supported collaborative learning. In: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, New York, NY, USA, ACM (2001) 24–31
2. Gutiérrez, S., Pardo, A., Delgado Kloos, C.: An adaptive tutoring system based on hierarchical graphs. In: Adaptive Hypermedia and Adaptive Web-Based Systems, Springer-Verlag (2004) 401–404

3. Valigiani, G., Biojout, R., Jamont, Y., Lutton, E., Republique, C.B., Collet, P.: Experimenting with a real-size man-hill to optimize pedagogical paths. In: SAC '05: Proceedings of the 2005 ACM Symposium on Applied computin g, New York, NY, USA, ACM Press (2005) 4–8
4. Murray, T.: Authoring knowledge-based tutors: Tools for content, instructional strategy, student model and interface design. Journal of the Learning Sciences **1** (1998) 5–64
5. Brusilovsky, P., Sosnovsky, S.: Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack. Journal of Educational Resources in Computing **5**(3) (2005)
6. Pardo, A.: A platform for parametrized exercises in web-based education. In: Int. Conf. of the Society for Information Technology and Teacher Education. (2002)
7. Ullrich, C., Chen, Q., Chen, S., Shen, L., Shen, R.: Investigating the reuse of course generation knowledge. In: Proc. of the 15th Int. Conf. on Computers in Education (ICCE2007), Hiroshima, Japan, IOS Press (2007)
8. Cristea, P., Tuduce, R.: Automatic generation of exercises for self-testing in adaptive e-learning systems: Exercises on ac circuits. In: 3rd Workshop on Adaptive and Adaptable Educational Hypermedia at the AIED'05 conference. (2005)
9. Tscherter, V.: Exorciser: Automatic Generation and Interactive Grading of Structured Exercises in the Theory of Computation (PhD Th.). Swiss Federal Institute of Technology (2004)
10. Anderson, J., Pelletier, R.: A development system for model-tracing tutors. In: Proc. of the International Conference on the Learning Sciences. (1991) 1–8
11. Murray, T.: Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education **10** (1999) 98–129
12. Azalov, P.: Synthetic exercises on the web. In: Int. Conf. on Information Technology: Coding and Computing (ITCC'05). (2005) 459–464
13. Fischer, S., Steinmetz, R.: Automatic creation of exercises in adaptive hypermedia learning systems. In: HYPERTEXT '00: Proceedings of the eleventh ACM on Hypertext and hypermedia, ACM (2000) 49–55
14. Lazarinis, F., Retalis, S.: Engineering an interoperable adaptive hypermedia testing tool supporting user adaptable strategies. In: IFIP Artificial Intelligence Applications and Innovations. 393–400
15. Mavrikis, M., Palomo, A.G.: Mathematical, interactive exercise generation from static documents. Electronic Notes in Theoretical Computer Science **93** (2004) 183–201
16. Gutiérrez, S., Pardo, A.: Sequencing in web-based education: Approaches, standards and future trends. In Jain, L.C., Tedman, R.A., Tedman, D.K., eds.: Evolution of Teaching and Learning Paradigms in Intelligent Environment, Springer (2007)
17. Wahlster, W., Kobsa, A.: User models in dialog systems. In Kobsa, A., Wahlster, W., eds.: User Models in Dialog Systems, Springer 4–34

18. Brusilovsky, P.: Adaptive and intelligent technologies for web-based eduction. KI **13**(4) (1999) 19–25
19. IMS Global Consortium: IMS question and test interoperability specification (QTI). In: http://www.imsglobal.org/question/index.html (last accessed January 2008)
20. Llorente, F.J.L.: ParEd: Web-based Parametric Exercise Editor (M.Th). University Carlos III of Madrid (2007)
21. Linillos, P.P., Gutiérrez, S., Pardo, A., Delgado Kloos, C.: Sequencing parametric exercises for an operating system course. In: IFIP Artificial Intelligence Applications and Innovation. (2006) 450–458