An Interactive Exercise Player for Math-Bridge

Johan Jeuring

Bastiaan Heeren

Technical Report UU-CS-2009-030

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

An Interactive Exercise Player for Math-Bridge

Johan Jeuring Bastiaan Heeren

School of Computer Science, Open Universiteit Nederland
P.O.Box 2960, 6401 DL Heerlen, The Netherlands
{johanj@cs.uu.nl, bhr@ou.nl}

Abstract

Math-Bridge is a European project which aims to provide facilities for bridging the mathematics gap between schools and higher education in Europe. The Open Universiteit Nederland is responsible for the interactive exercise player for Math-Bridge. This paper discusses the various forms interactions can take when solving mathematical exercises, the kind of feedback an exercise player should give according to teachers and developers of learning environments, and how strategies can be used to automatically calculate many of these kinds of feedback. Furthermore, it discusses some of the peculiarities of mathematical exercises that challenge our strategy framework.

1 Introduction

In most European countries there is a high demand for tailored remedial teaching materials for mathematics enabling the transition of students from schools to higher education, in particular engineering students. As a rule, existing content for remedial mathematics is available in a single language only, rarely online, and badly accessible. Moreover, it is represented in multiple formats, in various notations, and cannot be tailored to the learners' needs.

Math-Bridge aims at changing this situation to the better and helps to bridge the gap between schools and higher education in Europe. It will provide multilingual and multi-cultural semantic access (e.g. search and course generation) to remedial mathematics content which adapts to the requirements of a learner and his/her subject of study. It will bring together content from different European sources and offer it in a unified way. This access will be provided through a sustainable Pan-European learning service for remedial mathematics, which will be built by collecting appropriate learning resources, extending them in terms of structure and multi-linguality, and making them useful and easy-to-find. The extended formats of the content will make a wider use of standards and, hence, will make this content re-usable and "transferable" between different learning environments. In order to achieve its goals, Math-Bridge will study the (target) competencies required for target subjects of study, adapt existing semantic and multi-lingual search software, tailor assessment tools and methodologies, and adjust the cutting-edge ActiveMath learning environment to the remedy-scenario which includes specific diagnostic means and decisions for the transition from school to higher education. The service will be able to adapt to the level of learner competences and interests.

Moreover, Math-Bridge will enable collaborative authoring of the content on the basis of Creative Commons' licenses and improve instrumental support to collaborative authoring. This will stimulate collaborative production and assembly of educational content, which, we believe, is a future must. The results will be usable way beyond mathematics.

In this paper we will discuss one of the main tasks for which the Open Universiteit is responsible within the Math-Bridge project: the interactions in the exercise player, assessment tools, and diagnosis tools. We will use strategies for classes of exercises to give hints, worked-out examples, and to provide more detailed feedback. Mathematical exercises pose some challenges to our strategy framework, and we will describe some peculiarities of mathematical exercises.

This paper is organised as follows. Section 2 discusses the kind of interactive exercises in mathematical learning environments, and the kinds of feedback that are requested by teachers and learning environment developers. Section 3 introduces feedback services and strategies, and shows how they can be used to support interactions and feedback in learning environments. Section 4 discusses some of the kinds of mathematical exercises that are harder to model in our strategy framework. Section 5 concludes and gives future work.

2 An interactive exercise player

Learning mathematics requires practicing with the material that has to be mastered. Mathematics courses usually come with lots of exercises. Any remedial learning service for mathematics has to offer the possibility to practice with exercises. Furthermore, such a service should give feedback to students about their progress and errors. Many learning environments for mathematics offer interactive exercises to the user, and so does ActiveMath, the mathematics learning environment used in Math-Bridge (Goguadze, González Palomo, & Melis, 2005). Interactivity appears in various forms. Furthermore, when discussing feedback in interactive exercises with high-school teachers, university teachers, educational experts, and learning environment developers, we obtained requests for various forms of feedback. In this section we discuss the various kinds of interactivity, and the kind of feedback teachers expect.

We illustrate the various forms of interactions together with the corresponding feedback with an exercise about solving the quadratic equation $x^2 - 4x = 12$. Three possible derivations for this equation are shown in Figure 1, which will be our running example in this paper.

$$\begin{array}{l} x^2 - 4x = 12 \\ x^2 - 4x - 12 = 0 \\ (x-6)(x+2) = 0 \\ x = 6 \lor x = -2 \end{array}$$

$$\begin{array}{l} x^2 - 4x = 12 \\ x^2 - 4x + 4 = 16 \\ (x-2)^2 = 4^2 \\ x - 2 = 4 \lor x - 2 = -4 \\ x = 6 \lor x = -2 \end{array}$$

$$\begin{array}{l} x^2 - 4x = 12 \\ x^2 - 4x - 12 = 0 \\ \left| \begin{array}{l} D = (-4)^2 - 4 \cdot 1 \cdot -12 \\ \quad = 64 \\ \sqrt{D} = \sqrt{64} = 8 \end{array} \right. \\ x = \frac{4+8}{2} \lor x = \frac{4-8}{2} \\ x = 6 \lor x = -2 \end{array}$$

Figure 1: Three possible derivations for a quadratic equation

Multiple-choice questions. The most basic form of interaction is via multiple-choice questions. For example, four possible answers offered to our running example might be $x = 6 \lor x = -2$, $x = -6 \lor x = 2$, $x = 4 \lor x = -3$, and $x = 6$. With the wrong answers a teacher can store the common misconception that leads to this answer, and show this to a student who submits a wrong answer. It is labour-intensive to specify wrong answers for each multiple-choice question. Randomising these questions is desirable. Then we want to automatically calculate not just the correct answer, but also wrong alternatives that are based on common misconceptions.

Submitting final answers. Many exercises just ask for a final answer to a question. Checking whether or not a final answer is correct often involves more than a syntactic check: answers can be given in many different formats, and the students' answer has to be normalized to some extent to be able to verify correctness (Sangwin & Grove, 2006; Bradford, Davenport, & Sangwin, 2009; Heck & Gastel, 2006). An obviously correct variant that appears in solutions to quadratic equations is the order in which the solutions for x are given. It is not always easy to specify how much a student should simplify. For example, many teachers will want to see $2\sqrt{2}$ instead of $\frac{1}{2}\sqrt{32}$ in an answer, but sometimes the difference doesn't matter. Teachers want to specify erroneous answers together with appropriate feedback with an exercise, and show this to a student upon an error. Another response teachers wish to be able to give is a simpler question, in which a student only solves an initial part of the exercise. For example, if a student answers $x = 4 \lor x = -3$ to our running example, we might ask the question: Bring all terms to the left-hand side of the equation.

Solving exercises stepwise. Using pen-and-paper, students solve mathematical exercises step by step. ActiveMath can be used to mimic this process by offering interactions in which a student stepwise solves an exercise. The preferred way of interacting varies among teachers and learning environment developers: some prefer a student to select part of the current expression and a rule, and then apply the rule to the selection automatically (Beeson, 1998), others just want a student to submit a new expression (Chaachoua, Nicaud, Bronner, & Bouhineau, 2004), and yet others let a student both select a rule, and apply the rule to obtain the next expression (Boon & Drijvers, 2005). Various kinds of feedback are desirable.

- Is the submitted expression the final answer to the exercise? As in the case of submitting final answers, questions about simplification play a role here.
- Is the submitted expression similar to the previous expression? This implies that the student didn't take a step towards the solution, but instead performed some simplifications to the current expression.
- Is the submitted expression semantically equivalent to the previous expression? If not, the student has made a mistake.
- Does the submitted expression follow the strategy for solving the class of exercises? For instance, in our running example rewriting the left-hand side of the equation into $x(x-4)$ does not bring you closer to a solution. Hence, this step is not part of the strategy.
- Can the submitted expression be obtained by applying a common misconception to the previous expression? A common misconception for our running example would be to forget to change the sign when bringing the constant 12 to the left-hand side.

Furthermore, teachers want to be able to give hints about which step to apply next, to show how much progress a student has made towards a solution, or to show the complete derivation of the solution to the exercise. Note that when giving hints we can show only the step an expert would take towards the solution, or we can give all rules which bring the current expression closer to a solution at this point, including rules that lead to longer derivations. For example, after moving 12 to the left in our running example, we can choose to suggest just factorising the expression (the expert step), or also show the quadratic formula (allowing longer derivations).

Exercise completion. A good way to learn algebraic skills is to first study a worked-out example, than fill out a worked-out example from which some steps have been omitted, and only then completely solve an exercise (Sweller, Merriënboer, & Paas, 1998). ActiveMath offers fill-in-blanks exercises, which can be used for this purpose.

3 Feedback services

How do we realise the various kinds of feedback discussed in the previous section? To automatically calculate various kinds of feedback, we have introduced the concept of rewrite strategies for specifying exercises (Heeren, Jeuring, van Leeuwen, & Gerdes, 2008; Heeren, Jeuring, & Gerdes, 2010). A rewrite strategy specifies how an exercise is solved stepwise. For example, to solve a quadratic equation $ax^2 + bx + c = 0$, we first check if one of the simpler cases applies, in which either b or c equals 0. If not, we determine whether or not there exist "nice" factors that can be used to factorize the expression. This is the case in our running example. After factorizing, the resulting two linear equations are solved. If no nice factors are found, the quadratic formula is applied, and the two resulting answers are simplified.

A rewrite strategy is specified as a context-free grammar over rewrite rules, where the language for context-free grammars is extended with some constructs necessary for specifying exercises. A sequence of rewrite steps is a sentence of this grammar, if it follows the strategy. Correctness of a sequence of rewrite steps can be determined by parsing the sequence against the grammar. Our rewrite strategy language is a domain-specific language for specifying domain reasoners (Zinn, 2006).

Viewing a strategy as a grammar, and solving an exercise as constructing a sentence of the grammar has turned out to be a very useful way for automatically calculating various kinds of feedback. Strategies are used for calculating all the kinds of feedback described in the previous section.

We offer the various kinds of feedback as *services* (Gerdes, Heeren, Jeuring, & Stuurman, 2008). An exercise player uses our services to obtain feedback on a particular submission. Most services expect an expression, usually specified in some standard format such as OpenMath (The OpenMath Society, 2006), a strategy with which the exercise is solved, and a location in the strategy specifying which steps of the strategy have already been performed. The service then calculates the desired feedback and some other information, such as an updated location, which is returned to the exercise player. The updated information can be used in the next service request.

4 Services for mathematics

The strength of our approach based on rewrite strategies and services is that it is completely independent of the domain on which the rewriting takes place. We have used it successfully to solve particular classes of exercises in logic, relation algebra, and linear algebra. The exercises in mathematics we have worked on thus far introduced new challenges that need to be addressed. Most of these challenges followed from requests made by teachers.

Intermediate values. For some classes of exercises it is important to see and manipulate intermediate values. Applying the quadratic formula provides an example. Using this formula involves the following steps: identify the values for a, b, and c (the variables appearing in the quadratic formula), determine the discriminant, and in case the discriminant is positive, calculate its square root. Some of these steps are also visible in the right-most derivation in Figure 1. Omitting these intermediate values would make it hard to follow the calculation in a worked-out example, and would make the application of the quadratic formula quite involved.

Having intermediate values in a derivation is challenging because its associated steps are not rewrite rules. It is more like having a scribbling pad at ones disposal. From the perspective of a student, supplying intermediate values can be helpful since it provides guidance in performing a complex step. The extra steps also allows for new kinds of feedback: submitted intermediate values can be checked for correctness, and common misconceptions can be recognized and acted upon.

We have introduced a so-called *clipboard* to deal with this issue. This clipboard is communicated as part of the context that is attached to the current expression. Values can be written to (and read from) the clipboard, both by the exercise player and the domain reasoner. Hence, a common understanding of the content of the clipboard is needed at both ends.

Rounding numerical values. Although most interactive exercises require students to provide an exact answer to a question, in certain cases it is desirable to ask for an approximation of the final answer. For example, instead of accepting $x = \frac{1}{2}\sqrt{17}$ as a solution, an environment could make a student use a pocket calculator and submit $x \approx 2.062$.

Approximations complicate diagnosing intermediate student answers: rounding errors are propagated, and comparing floating-point numbers is notoriously difficult. To circumvent these problems, we only accept approximations as a final step, and we make the rounding explicit by choosing a different symbol (that is, $\approx$).

Exercise in parts. Some exercises are solved in parts. For example, a standard strategy for solving a quadratic inequation (such as $x^2 - 4x < 12$) is to first solve it as an equation, and then use the result to provide an overall answer. Simply turning the inequation into an equation as a first step would be inappropriate because these two clearly have different meanings and solutions.

A solution is to place the inequation in the context of the current expression at the introduction of the equation. This way, a consistent meaning is available during the whole derivation.

Implicit simplification. The treatment of automatic simplification and the use of canonical forms in the domain of mathematics is particularly subtle. For instance, applying the distribution rule $a(b + c) \rightsquigarrow ab + ac$ to the expression $4(x+2)$ would result in $4x+8$, thus performing the simplification step $4 \cdot 2 \rightsquigarrow 8$ silently. This makes perfect sense for an interactive exercise that focuses on solving equations, and not on performing basic calculations. The degree to which expressions are simplified automatically can vary between exercises.

In the case of an exercise on solving quadratic equations, one must decide on how to simplify square roots. For this, we use views (Heeren & Jeuring, 2009) in our domain reasoners. A view describes a canonical form. For instance, we could choose to view $\frac{\sqrt{32}}{2}$ in the canonical form $2\sqrt{2}$, and to not distinguish between the two expressions. Both expressions are equally well-suited as a final answer. Alternatively, we could decide to make a distinction, and to require an extra step by the student to turn $\frac{\sqrt{32}}{2}$ into $2\sqrt{2}$ if the latter is expected as the final answer. Regardless of whether the simplification of square roots is implicit or explicit, an environment should be able to provide useful feedback to the student.

With respect to simplifying square roots, one should take into account that not all square roots can be normalized easily without the help of a pocket calculator, especially when large numbers occur. Similarly, finding the factors of a quadratic equation can be of varying difficulty. Interactive exercise assistants should allow teachers to indicate the boundaries of what can be expected from a student, and what not.

5 Conclusion

To bridge the gap in mathematical competencies between schools and higher education, the European Math-Bridge project provides on-line mathematics learning facilities. The Open Universiteit is responsible for the interactive exercise player for Math-Bridge. In this paper we have discussed the various forms interactions can take, and the kinds of feedback desired by teachers and learning environment developers. Strategies can be used to automatically calculate all these kinds of feedback. We have discussed some of the peculiarities of mathematical exercises that pose challenges to our strategies framework.

In the near future we will develop feedback services for the domains that are necessary for courses developed within Math-Bridge. We expect to develop many domains and rewrite strategies. We will investigate how these domains, rewrite rules, and strategies are best organised, maintained, and reused.

References

Beeson, M. J. (1998). Design principles of Mathpert: Software to support education in algebra and calculus. In N. Kajler (Ed.), *Computer-human interaction in symbolic computation* (pp. 89–115). Springer-Verlag.

Boon, P., & Drijvers, P. (2005). *Algebra en applets, leren en onderwijzen (algebra and applets, learning and teaching, in Dutch).* (`http://www.fi.uu.nl/publicaties/literatuur/6571.pdf`)

Bradford, R., Davenport, J. H., & Sangwin, C. J. (2009). A comparison of equality in computer algebra and correctness in mathematical pedagogy. In *Calculemus '09/mkm '09: Proceedings of the 16th symposium, 8th international conference. held as part of cicm '09 on intelligent computer mathematics* (pp. 75–89). Berlin, Heidelberg: Springer-Verlag.

Chaachoua, H., Nicaud, J.-F., Bronner, A., & Bouhineau, D. (2004). Aplusix, a learning environment for algebra, actual use and benefits. In *Icme 10: 10th international congress on mathematical education.* (Retrieved from `http://www.itd.cnr.it/telma/papers.php`, May 2008)

Gerdes, A., Heeren, B., Jeuring, J., & Stuurman, S. (2008). Feedback Services for Exercise Assistants. In D. Remenyi (Ed.), *The proceedings of the 7th european conference on e-learning* (p. 402-410). Academic Publishing Limited.

Goguadze, G., González Palomo, A., & Melis, E. (2005). Interactivity of exercises in ActiveMath. In *International conference on computers in education, icce 2005.*

Heck, A., & Gastel, L. v. (2006). Diagnostic testing with Maple T.A. In M. Seppälä, S. Xambo, & O. Caprotti (Eds.), *Proceedings of the Web Advanced Learning Conference and Exhibition, WebALT 2006* (pp. 37–52). Oy WebALT Inc.

Heeren, B., & Jeuring, J. (2009). Canonical forms in interactive exercise assistants. In L. Dixon et al. (Ed.), *Mkm 2009: Mathematical knowledge management, 8th international conference* (Vol. 5625, pp. 325–340). Springer-Verlag.

Heeren, B., Jeuring, J., & Gerdes, A. (2010). Specifying rewrite strategies for interactive exercises. *Mathematics in Computer Science*. (To appear. This is an extended version of (Heeren et al., 2008))

Heeren, B., Jeuring, J., van Leeuwen, A., & Gerdes, A. (2008). Specifying strategies for exercises. In S. Autexier et al. (Ed.), *Mkm 2008: Mathematical knowledge management, 7th international conference* (Vol. 5144, pp. 430–445). Springer-Verlag.

Sangwin, C. J., & Grove, M. (2006). Stack: addressing the needs of the ¨neglected learners¨. In M. Seppälä, S. Xambo, & O. Caprotti (Eds.), *Proceedings of the Web Advanced Learning Conference and Exhibition, WebALT 2006* (pp. 81–96). Oy WebALT Inc.

Sweller, J., Merriënboer, J. v., & Paas, F. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, *10*, 251–295.

The OpenMath Society. (2006). *The OpenMath Standard.* (`http://www.openmath.org/standard/index.html`)

Zinn, C. (2006). Supporting tutorial feedback to student help requests and errors in symbolic differentiation. In M. Ikeda, K. Ashley, & T.-W. Chan (Eds.), *Its 2006* (Vol. 4053, pp. 349–359). Springer-Verlag.

Acknowledgments

This work was made possible by the Math-Bridge project of the Community programme eContent*plus*, and by the SURF Foundation, the higher education and research partnership organization for Information and Communications Technology (ICT). The paper does not represent the opinion of the Community, and the Community is not responsible for any use that might be made of information contained in this paper. For more information about SURF, please visit `http://www.surf.nl`. We have discussed our feedback services extensively with Peter Boon.