

# Design Patterns for the Use of Technology in Introductory Mathematics Tutorials

Christine Bescherer and Christian Spannagel

University of Education Ludwigsburg, Reuteallee 46, 71634 Ludwigsburg, Germany  
bescherer@ph-ludwigsburg.de, spannagel@ph-ludwigsburg.de

**Abstract.** Learning mathematics actively, oriented at mathematical processes, in a technology-enhanced learning environment differs widely from learning in traditional mathematics courses. The traditional lecture – an expert presents the knowledge the learners have to acquire – usually doesn't activate the students' thinking. This article introduces three didactical design patterns which describe how university students can be enabled to use technology in order to explore and solve mathematical problems in open learning scenarios: TECHNOLOGY ON DEMAND, HELP ON DEMAND, and FEEDBACK ON DEMAND.

**Keywords:** Integration of ICT, Open Flexible Learning, Problem Solving, Teacher Education, Teaching Methods, Assessment, Didactics, Higher Education, Learner-Centred Learning, Learning Mathematics.

## 1 Introduction

Teaching mathematics is often understood as presenting solutions to given problems. In universities, many introductory mathematics tutorials which accompany lectures are exactly designed in this way: experts (professors, lecturers or tutors) develop solutions on the chalkboard [1]. Students follow the demonstrations and take notes or – more precisely – try to follow the lecture. This approach to mathematics teaching does not foster active thinking and problem solving. Students at least in the first semesters are encouraged to copy standard solutions and not to develop creative strategies by themselves. In addition, students do not communicate and collaborate – they just listen.

Bescherer, Spannagel, and Müller [2] introduced a pattern for introductory mathematics tutorials in order to foster active, collaborative, and process-oriented mathematics learning: ACTIVATING STUDENTS IN INTRODUCTORY MATHEMATICS TUTORIALS (*Pattern for introductory mathematics tutorials following a constructivist approach*). Students work in small groups (three or four persons) on complex problems supported by tutors. Every week, a set of 5 to 6 new problems is given via a learning management system (LMS). Students are allowed to choose the problems they want to work on. They are encouraged to work on the problems in groups during the tutorial session or at other times.

Tutors do not give “the correct” solutions. They guide the students doing mathematics by asking the right questions and giving hints on strategies. Furthermore, students are allowed to use every resource they want: they may use books, the internet or computer tools like spreadsheet calculators or dynamic geometry systems (DGS). Especially the use of technology requires the learning scenario to be designed in a special way so that the IT is used as a thinking tool and a medium for exploration and not just to do the complicated math.

In section 2 three aspects using technology in introductory mathematics tutorials are described: TECHNOLOGY ON DEMAND, HELP ON DEMAND, and FEEDBACK ON DEMAND. The structured description format of didactical design patterns is used to support the understanding and the reusability.

The idea of design patterns to describe working solutions to recurring problems originates from architecture [3] and software design [4]. To use design patterns for describing pedagogical problem solutions is not a new idea. On the site [www.pedagogicalpatterns.org](http://www.pedagogicalpatterns.org) there are a lot of patterns available. However they describe the pedagogical problems in a very broad way and have to be adapted quite extensively to i.e. introductory mathematics tutorials at university level. Therefore we think that pedagogical decisions in teaching subject matters always depend – among other things – on the subject immanent particularities. That is also why we call our patterns ‘didactical design patterns’ following the German idea of ‘didactics’ as the science of learning and teaching a specific subject i.e. didactics of mathematics or didactics of foreign languages. ‘Pedagogical’ in our view is not related to a specific subject.

The first and the second of our patterns focus on how technology can be integrated into introductory tutorials where students work actively on complex math problems.<sup>1</sup> The third pattern deals with the assessment of the students’ work in order to give them informative feedback on their mathematical processes. Conclusions and remarks about future work are explained in section 3.

## 2 Three Patterns for Technology-Enhanced Mathematics Learning

### 2.1 The TECHNOLOGY ON DEMAND Pattern

#### 2.1.1 Problem / Challenges / Motivation

Students solving math problems should learn *when* to use *which* software in *which* context. They should be able to use the software whenever they think it will be useful. Students should experience the usefulness of technology solving mathematical problems and reflect on it.

#### 2.1.2 Forces

- To solve math problems with IT is not always necessary so there has to be created a demand for technology to solve the problems.

---

<sup>1</sup> Of course technology is not always necessary to solve mathematical problems but there are enough examples of traditional math tutorials so we don’t follow this line of discussion.

- Students have no or low abilities in using related software in itself so they avoid using IT as long as possible.
- Exploring mathematical assumptions using technology requires i.e. systematic approaches of varying specific parameters and keeping track of the changes which are not normally known to students from school.

### 2.1.3 Solution

Math problems are selected where the use of technology is necessary. There are several ways to achieve this: (1) Problems must be of sufficient high complexity. (2) Problems must induce operations which would be too much work or too monotonous work to be done by hand (for example, always the same calculations with different data). (3) Representations or visualizations of data which can't be created efficiently without the software are needed to solve the problems.

Students get instructions how to use the specific software (s. section 2.2). Math problems should contain hints on which software is appropriate. These hints should be as open as possible. For example, they should not mention specific software packages (as Open Office Calc), but types of software (as spreadsheet calculation programs). Ideally, alternatives are given in a way that students can reflect on the advantages and disadvantages of different tools in the specific context. In addition, students must have access to computer tools whenever they need it.

The phrasing of the problems includes questions, tasks and hints which guide the exploring using the technology without suppressing the possibility to follow own ideas and other paths.

### 2.1.4 Rationale

The learning of software usage is most effective in contexts where it is necessary to use the software. The need to use a software tool should come before the instruction not the other way around. This is called *just-in-time learning* or *learning-on-demand* ([5],[6],[7]).

Computer applications are cognitive tools when they support people's thinking. "Cognitive tools refer to technologies, tangible or intangible, that enhance the cognitive powers of human beings during thinking, problem solving, and learning." ([8], p693). Cognitive tools allow for creating useful representations, they help to explore a given situation in microworlds, they support deep thinking about content, or they just take away routine jobs from the learner to free cognitive resources [9].

Cognitive tools in the context of learning mathematics can be spreadsheet calculation programs, dynamic geometry systems, computer algebra systems, or simply handheld calculators. The problems have to be posed so that the software i.e. allows to explore an assumption or to falsify the obvious first idea to a solution. If the software can be simply used to avoid thinking then the problems has to be changed.

### 2.1.5 Examples

A typical geometry problem inducing the need for technology is the following:

*Given the instruction of inverting points with respect to an inversion circle (without giving away the mathematical term), explore the following questions:*

- *What is the inverse of a line?*
- *What is the inverse of a circle?*
- *What happens if the circle is moved?*

*Hints/Techniques:*

- *You can use a dynamic geometry system for exploration.*

A similar example in the field of algebra (cf. [2]):

*Make conjectures of several unit fractions concerning their decimal representation.*

- *What kind of decimal do you get?*
- *If it is not a terminating decimal: How long are the periods and the delays of the periods? Make conjectures on the base of your data.*
- *Which properties determinate the kind of decimal? Which properties determinate the length of the period and the delay?*
- *Test your hypotheses with other unit fractions.*

*Hints/Techniques:*

- *You can use the spreadsheets made available in our LMS.*
- *Which of the unit fractions are good indicators for your conjectures?*

### **2.1.6 Related Patterns<sup>2</sup>**

Activating students in introductory mathematics tutorials, help on demand, feedback on demand

## **2.2 The HELP ON DEMAND Pattern**

### **2.2.1 Problem / Challenges / Motivation**

Students differ in their computer fluency and in their previous knowledge regarding software used in the math tutorials. All students should be enabled to use the tools for problem solving without spending too much effort in learning the tools.

### **2.2.2 Forces**

Often the software is taught before students really need it. This may result in lengthy demonstrations of complex procedures. But normally some students already know how to use the tools. For those the demonstration is unnecessary and boring. For novices, it can be too much information in advance, and afterwards they don't remember what to do.

### **2.2.3 Solution**

Help on technology must be right at hand *when the information is needed*. In the first place, peer support should be fostered. Students can help each other when working together at the same computer. Tutors may also give hints on how to use the software.

In cases where students use the software alone at home, or where all students are likely to be novices regarding a specific tool, instructions should be given where the

---

<sup>2</sup> This section lists all related patterns and is also an inherent part of a didactical design pattern.

core procedures are described. Procedures explained should be analogue to those needed by the students and not step-by-step explanations of how to solve the given problem. The worked example should be a mathematically simpler example where all the essential procedures are mentioned that can be used in the current problem situation as well. This approach is part of LoDiCs – methodical structures used to learn IT use and fundamental principles of information technology while working on subject matter problems – which were introduced by Bescherer [6].

Instructions for using software can be text manuals, interactive worksheets or screen videos. These can be provided in the LMS for download. Thus, students can access them on demand.

#### 2.2.4 Rationale

The use of an analogue example to introduce all the necessary software features for solving a specific mathematical problem is derived from the modelling and scaffolding parts of the cognitive apprenticeship model introduced by Collins, Brown, and Newman [10] and discussed in detail in Bescherer [6].

Users of software applications do not read the documentation in advance; they access it when they face a problem ([11], [12]). They need *just-in-time help*. Although integrated help is often ignored by users, research shows that accessing on-demand help may lead to better learning outcomes [13].

Manuals should be modular, task-oriented, contain as little text as possible, and should include information about error-recovery ([14], [15]). In addition, screenshots may help to build mental models and to identify interface elements ([16]).

Screen videos should be accompanied by simultaneous spoken text (*modality principle* and *temporal contiguity principle*; [17]). Although initial research regarding the effects of animated demonstrations was disillusioning [18], there is evidence that carefully designed screen videos may lead to a better performance compared to text manuals ([19], [7]).

#### 2.2.5 Examples

Given the geometry example of section 2.1.5, screen videos can be produced which show the basic procedures needed to explore the *inversion with respect to an inversion circle*: how to draw a circle, and how to draw a perpendicular line.

The algebra example of section 2.1.5 is part of the LoDiC ‘Decimal Fractions’ which can be found online at [www.lodics.de](http://www.lodics.de).

#### 2.2.6 Related Patterns

Activating students in introductory mathematics tutorials, technology on demand, feedback on demand

### 2.3 The FEEDBACK ON DEMAND Pattern

#### 2.3.1 Problem / Challenges / Motivation

To support learning processes informative feedback shouldn’t be given only on the resulting products i.e. the mathematical solution but also on the mathematical processes like algebraic transformations, comparing different sequences of solution steps up to higher level processes like reasoning, representing, problem solving, and the

learning process itself. This is a challenge not only in large introductory lectures with many participants but also in small tutorial groups, because tutors cannot monitor all processes performed by 10 or 20 students simultaneously. Students should be able to get process-oriented feedback when they need it (*demand claimed by the student*), and lecturers should be able to select interesting (correct or wrong) mathematical processes for discussion in the lecture sessions (*demand claimed by the professor or lecturer*).

### 2.3.2 Forces

Students need feedback on their mathematical processes in order to improve them and on their learning process to know ‘where they stand’. Tutors can only monitor the process of one student or one team of students at the same time. Feedback on mathematical processes can be complex and time-consuming; especially learning processes must be observed over some time. Once a process is diagnosed, feedback must be carefully chosen to be informative, encouraging, and not intimidating.

### 2.3.3 Solution

Processes are recorded and analyzed by specific software tools. Normally there are many different processes involved leading to the correct solution of a complex mathematical problem. And there are even more ‘wrong paths’ followed in solving mathematical problems by students. It would be impossible to implement algorithms which record and classify *all* processes correctly. Instead it is often sufficient to detect standard solutions and standard mistakes. Processes which cannot be automatically categorized can then be forwarded to the tutor or lecturer to be assessed (*semi-automated assessment*, [20]). This means:

- There must be tools which are able to record and analyze processes performed in applications like DGS or spreadsheet calculators. These can be generic tools which are able to analyze user-program interactions (like *Jacareto*; [21]), or analysis features directly implemented in the applications (like the theorem checking technique in *Cinderella*; [22]).
- Students must be able to ask for process-oriented feedback, for example by clicking on a “feedback”-button. If the analysis tool is able to classify the process, feedback can directly be given. Otherwise, the record of the process can be send to the tutor or lecturer for assessment.
- The lecturer must be able to browse the recorded processes and select interesting (correct or false) ones to be presented in the next lecture session. Thus, there must be a repository of processes submitted by the students (cf.[23]). In addition, process records must be stored in a manner that they can be replayed for demonstration. This can be done with capture & replay tools like *Jacareto* [24].

### 2.3.4 Rationale

To give adequate feedback, theory on feedback and assessment must be considered (i.e. [25]). Assessment is the “systematic evaluative appraisal of an individual’s ability and performance in a particular environment or context” ([26], p. 474). It is normally based on artifacts (papers, written tests, portfolios, ...) as well as teacher observations, oral

contributions, or learners' presentations. It becomes *formative assessment* when it is used to improve teaching and learning which means "the evidence is actually used to adapt the teaching work to meet learning needs" ([27], p.10).

Regarding assessment in learning mathematics there is a worldwide discussion going on that changes in teaching mathematics are connected strongly to changes in assessment of mathematical knowledge. The *Assessment Standards for School* were published by the National Council of Teachers of Mathematics in 1995 with the emphasis on *assessing student's full mathematical power* instead of *assessing only students knowledge of specific facts and isolated skills*, or the demand to regard assessment as something continual and recursive and not something sporadic and conclusive ([28], p.83).

It is important that process-oriented feedback is informative and does not create negative emotions. It should strengthen the perceived competence of the students which is an important factor for motivation [29].

Semi-automated assessment reduces the claim to automatically interpret *all* processes. Instead, it combines the ability of computers to detect standard solutions or mistakes and the professional skills of the tutors and lectures to understand exceptional solutions. For further background on intelligent assessment where semi-automatic assessment is a part of see [20].

### 2.3.5 Examples

The three year project SAiL-M (semi-automated assessment of individual learning processes in mathematics), started at the end of 2008 and funded by the German Federal Ministry of Education and Research is working on implementing just this kind of semi-automatic feedback. We are currently working on several examples which will be implemented in the summer term 2009.

In the algebra example of section 2.1.5 a checking-system can be developed which checks whether a fraction is terminating or periodic with or without delay and gives feedback on the learners' assumptions. This can be done with an underlying computer algebra system and would be an example for feedback claimed by the learner.

Process-oriented feedback can be given if the student hasn't checked out all types of fractions. Then it can be assumed that he can not form a hypothesis yet, the feedback will be something like "There are still more types of fractions. Check them out before you hypothesize."

### 2.3.6 Related Patterns

Activating students in introductory mathematics tutorials, technology on demand, help on demand

## 3 Conclusion and Future Research

The three patterns introduced here are dealing with aspects of active, technology-enhanced, and process-oriented learning in introductory mathematics tutorials. It has been emphasized that the problem should create a demand to use technology, the help and feedback should be provided on demand only. This follows the philosophy of open learning scenarios – a radical change in lecturers' as well as students' thinking about learning mathematics.

Process-oriented feedback is very task-specific and must be implemented for each class of problems separately. Therefore it is useful to have frameworks which facilitate the implementation of process-oriented feedback (cf. [21]). Concerning semi-automated assessment there are still many open research questions.

## Acknowledgements

Work described in this article is funded by the German Federal Ministry of Education and Research.

## References

1. Holton, D.: *The Teaching and Learning of Mathematics at University Level: An ICMI Study*. Kluwer Academic, Dordrecht (2001)
2. Bescherer, C., Spannagel, C., Müller, W.: Pattern for Introductory Mathematics Tutorials. In: *Proceedings of the EuroPLOP 2008 Conference* (in press, 2008)
3. Alexander, C., Ishikawa, S., Silverstein, M.: *A pattern language. Towns, buildings, construction*. Oxford U.P, New York (1977)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading (1995)
5. Eisenberg, M., Fischer, G.: Symposium: learning on demand. In: *Proceedings of the Fifteenth Annual Conference of the Cognitive Science*, pp. 180–186. Lawrence Erlbaum Associates, Hillsdale (1993)
6. Bescherer, C.: LoDiC – Learning on Demand in Computing. In: *Proceedings of 8th IFIP World Conference on Computers in Education 2005, Cape Town, July 4–7 (2005)*
7. Spannagel, C., Girwidz, R., Löthe, H., Zendler, A., Schroeder, U.: Animated Demonstrations and Training Wheels Interfaces in a Complex Learning Environment. *Interacting with Computers* 20(1), 97–111 (2008)
8. Jonassen, D.H., Reeves, T.C.: Learning with technology: Using computers as cognitive tools. In: Jonassen, D.H. (ed.) *Handbook of research for educational communications and technology*, pp. 693–719. Macmillan, New York (1996)
9. Salomon, G.: No distribution without individuals' cognition: a dynamic interactional view. In: Salomon, G. (ed.) *Distributed cognitions. Psychological and educational considerations*, pp. 111–138. Cambridge University Press, New York (1993)
10. Collins, A., Brown, J.S., Newman, S.E.: Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing, and Mathematics. In: Resnick, L.B. (ed.) *Knowing, Learning, and Instruction. Essays in Honor of Robert Glaser*, pp. 453–494. Erlbaum, Hillsdale (1989)
11. Penrose, J.M., Seiford, L.M.: Microcomputer user's preferences for software documentation: an analysis. *Journal of Technical Writing and Communication* 18(4), 355–366 (1988)
12. Ames, A.L.: Just what they need, just when they need it: an introduction to embedded assistance. In: Tilley, S. (ed.) *Communicating in the New Millennium. Proceedings of the 10<sup>th</sup> Annual International Conference on Systems Documentation*, pp. 111–115. ACM, New York (2001)
13. Alevén, V., Stahl, E., Schworm, S., Fischer, F., Wallace, R.: Help seeking and help design in interactive learning environments. *Review of Educational Research* 73(3), 277–320 (2003)



14. Lazonder, A.W., van der Meij, H.: The minimal manual: is less really more? *International Journal of Man-Machine Studies* 39, 729–752 (1993)
15. Albers, M.J.: Designing and writing to reduce user errors. In: *Proceedings of the 50<sup>th</sup> Annual Conference of the Society for Technical Communication*, pp. 286–288. STC, Arlington (2004)
16. Gellevij, M., van der Meij, H., de Jong, T., Pieters, J.: Multimodal versus unimodal instruction in a complex learning context. *The Journal of Experimental Education* 70(3), 215–239 (2002)
17. Mayer, R.E.: *Multimedia learning*. Cambridge University Press, New York (2001)
18. Palmiter, S., Elkerton, J.: Animated demonstrations for learning procedural computer-based tasks. *Human-Computer Interaction* 8, 193–216 (1993)
19. Atlas, R., Cornett, L., Lane, D.M., Napier, H.A.: The use of animation in software training: pitfalls and benefits. In: Quinones, M.A., Ehrenstein, A. (eds.) *Training for a rapidly changing workplace. Applications of psychological research*, pp. 281–302. American Psychological Association, Washington (1997)
20. Bescherer, C., Kortenkamp, U., Müller, W., Spannagel, C.: Intelligent Computer-Aided Assessment in Mathematics Classrooms. In: McDougall, A. (ed.) *Researching IT in Education: Theory, Practice and Future Directions*, Routledge (in press, 2009)
21. Spannagel, C., Gläser-Zikuda, M., Schroeder, U.: Application of Qualitative Content Analysis in User-Program Interaction Research. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* 6(2), Art. 29 (2005)
22. Richter-Gebert, J., Kortenkamp, U.: *The Interactive Geometry Software Cinderella*. Springer, Heidelberg (1999)
23. Oppermann, R., Thomas, C.G.: Learning and problem solving as an iterative process: Learner's Living Repository: LEAR. In: Stephanidis, C. (ed.) *Towards user interfaces for all: current efforts and future trends. Proceedings of the ERCIM Workshop*, Heraklion, Crete, Greece, October 30-31 (1995)
24. Schroeder, U., Spannagel, C.: Supporting the Active Learning Process. *International Journal on E-Learning* 5(2), 245–264 (2006)
25. Dempsey, J.V., Sales, G.C. (eds.): *Interactive Instruction and Feedback*. Educational Technology Publications, Englewood Cliffs (1993)
26. Payne, D.: *Applied Educational Assessment*. Wadworth Publishing Company, Belmont (1997)
27. Black, P., Harrison, C., Lee, C., Marshall, B., Wiliam, D.: Working Inside the Black Box: Assessment for Learning in the Classroom. *Phi Delta Kappan* 86(1), 9–21 (2004)
28. NCTM: *Assessment Standards for School Mathematics*, The National Council of Teachers of Mathematics, Reston VA (1995)
29. Deci, E.L., Ryan, R.M.: An overview of self-determination theory: an organismic-dialectical perspective. In: Deci, E.L., Ryan, R.M. (eds.) *Handbook of self-determination research*, pp. 3–33. The University of Rochester Press, Rochester (2002)