DIDACTICAL DESIGN PATTERNS FOR THE APPLICATIONS OF SOFTWARE TOOLS

Paul Libbrecht*, Marc Zimmermann**

*) Mathematics Didactics, Martin Luther University of Halle

**) University of Education Ludwigsburg

We report on didactical design patterns for mathematics education with a special focus on the impact on the software construction process. We describe how such short texts, in the first instance developed for educators, can also be used by developers of learning software. We claim that such an application supports a more effective usage of the tools, thus raises their quality.

The paper presents one of the design patterns in details, Representation-on-Demand, and illustrates its applications through several tools.

INTRODUCTION

Numerous software learning tools have appeared in the last ten years. Just as technology found its way into society (personal computers, mobile phones, tablets,...), technology found its way into the learning places. But what is the best way to learn with learning software? This question has not been solved till now. Learning software is not yet commonplace in daily school lessons or at university lectures, even though there are many well developed learning tools.

In the authors' experience, from the view of university education, many students still prefer working with pencil and paper. On the one hand using technology and learning tools is a new concept to the students. Before working on a task they have to learn to handle the program. Therefore it seems faster for them to get a solution by hand. On the other hand, students argue that in the end the exams also have to be done by hand. Using learning software makes no sense, because no matter what, you have to work on a task with pencil and paper to prepare for the exam. Additionally, students in teacher education argue that these programs are not going to be used in school, therefore there is no need to use them. As a result, for many students the learning tools do not connect to the learning behaviour.

Hence, learning tools, which we shall understand as any software that helps learning, have to be designed so that students will use them. They have to follow the learning behaviour of the students. We claim that learning tools could be enriched, by using the same patterns that are used by teachers and educators. In this paper, we propose the dissemination of short texts describing patterns in the learning processes.

The didactical design patterns are the focus of this paper. They form a collection of short and readable texts, which members outside the (mathematics) pedagogy community can exploit. We are particularly interested in the applications of the didactical design patterns to the learning tools' development and documentation: we observed that the language to describe the learning tools may lack the concepts of the mathematics pedagogy and may sometimes be far from the users' day to day tasks.

A common language is needed for the technology enhanced learning and the mathematics pedagogy research and practice communities to understand each other. This has been noted by Noss (2009) who stresses the importance of interdisciplinarity for the field of technology enhanced learning. A few initiatives are emerging to help summarizing the branch in the form of encyclopedic knowledge. Among others, on the computer science side, the TEL Thesaurus (http://tel-thesaurus.net/) and the interaction-design foundation (http://interaction-design.org/) are both present, and on the mathematics education side, one finds MaDiPedia (http://madipedia.de/) or the ReMath scenarios (http://remath.cti.gr/).

The didactical design patterns we present here are a form of common language with a different perspective than the works above: they depict recurrent schemes which are backed by literature in pedagogy. Their *usage* by software designers or teachers should be simple and backed by a sufficient vocabulary that enables ease of discussion. We attempt to document possible application processes in the software construction and documentation.

Our contribution connects the three themes of the working group: it contributes to the design and use of technologies by proposing patterns to raise the quality of the learning experience (theme 1.1). It also provides means to raise the impact of the use of the technologies in their learning (theme 2.1). Finally, it proposes patterns that support best practices in using the technologies (theme 3.3).

OUTLINE: This paper first introduces the principles behind didactical design patterns and the concepts they relate to. It then describes *Representation on Demand*, the central pattern of this paper. Two of its exemplary implementations are then described. In the conclusion, future research directions are outlined.

DIDACTICAL DESIGN PATTERNS

In general, design patterns were first described in the subject of architecture by Alexander (Alexander., Ishikawa, & Silverstein, 1977). Even there, they were used long times before. The issue of design patterns was to give a general solution for recurring problems when constructing buildings. This approach was adopted in the 90s by computer scientists (Gamma, Helm, Johnson, & Vlissides, 1995) for recurring problems they encountered when writing programs, e.g. algorithm or programme concepts, which are used in many programs. For these, the design patterns are still used today in these fields. Therefore, design patterns are used to describe possibilities to solve challenges [1] on pedagogical or didactical [2] problems. This does not mean, that they are an instruction that says, after doing this all is fine. They provide (theoretical grounded) hints and outlooks in which way the challenges can be solved.

To describe these kinds of challenges with a possible solution as a pattern in education is not new. The "Pedagogical Patterns Project" (Bergin et al., 2012; www.pedagogicalpatterns.org) contains a lot of very broad patterns relating to pedagogy. Vogel and Wippermann (2004) as well as Niegemann and Niegemann (2008) introduced didactical design patterns as a possibility to document didactical knowledge. The advantages of patterns are that they provide named short and repeatable approaches to a (pedagogical) problem. Thus, they are easy to read, understand, and thus to use by everyone, especially teachers to optimize learning scenarios and processes.

Figure 1 depicts how patterns are situated to other texts in terms of their applicability, didactical diversity (the diversity of situations it can be employed in), and their coverage (how many dimensions does it take in account).

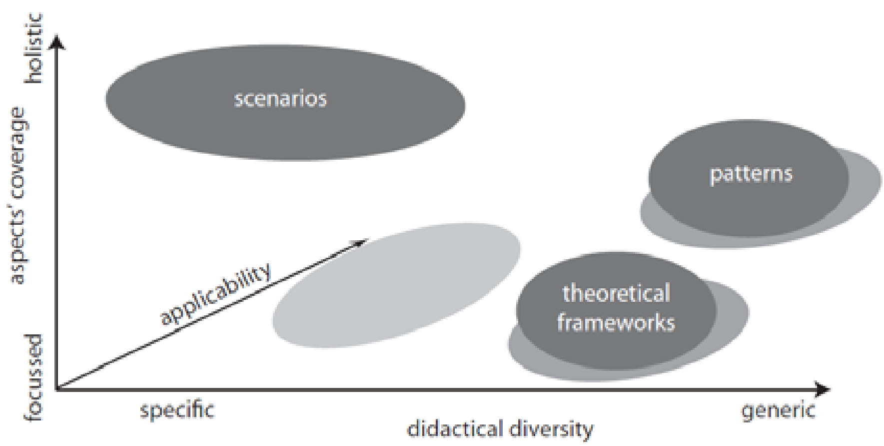

Figure 1: Localisation of didactical design patterns, compared to learning scenarios.

In contrast to the pedagogical patterns (c.f. Bergin et al., 2012), which describe challenges in learning scenarios in a very broad way, e.g. students should be active while learning (p. 17, ibid.), didactical design patterns described in this paper are somewhat more precise to the learning process. Didactical design patterns differentiate themselves from learning scenarios in that they are abstract and describe a more general situation. They more focus on didactical principles in learning scenarios, e.g. when does a learner need to get a hint on a problem (Zimmermann, Herding, & Bescherer, 2013). The patterns are the work of the SAiL-M project (Semi-automatic Analysis of individual Learning-processes in Mathematics). They all are available on the web at www.sail-m.de and at sail-m.i2geo.net. The availability of their text as simple web-pages with a short title, a direct URL, and under an open content license makes them considerably easier to mention in electronic communication, a fact of growing importance.

There are different and various styles to describe didactical design patterns the patterns developed in the project (Bescherer & Spannagel, 2009; Bescherer, Spannagel, & Müller, 2008) follow a structure made up of the following ingredients:

- **challenge/motivation** (problem): The issue intended to be solved is introduced.
- **forces**: Factors influencing the described problem and, therefore, no easy solution is possible.
- **solution**: One (general) recommendation is formulated.
- **rationale**: Theoretical reasoning on which the possible solution is based on. It helps readers to dig out justifications of particular aspects.
- **examples**: Precise situations where the pattern has been successfully used.
- **related patterns**: Connection to patterns that are relevant when applying this one.

DIDACTICAL DESIGN PATTERNS IN SOFTWARE DEVELOPMENT

As well as in school, didactical design patterns can be used in the develoment process of learning tools. Didactical design patterns can be interesting because they represents a compact set of guidelines for several types of applications, which focuses on the essential aspects of the learning processes. Software developers are left free to use them when creating a tool but should always be aware of an environment where the pattern can take place.

For example it is important in a learning process to get hints and/or feedback and learning tools often provide them. But the hints that are given by the tools should not reveal the whole solution at once. They have to be given in different levels of detail and have to be individual to the learner's problems. This is an approach a human tutor normally applies when helping students working on tasks. Hence learning tools should also provide hints and feedback at different levels and provide them when the learner demands it. The Hint on Demand Pattern (Zimmermann, Herding, & Bescherer, 2013) describes how this didactical approach can be implemented in learning tools.

Another question is, when the help or technical support should be given to the learner. Giving the help beforehand, because there are problems where learners normally get stuck, is disadvantageous, just as giving it in the time a mistake occurs. The Help on Demand pattern (Bescherer & Spannagel, 2009) describes that help should be given just in time and when the learner actively demands it.

THE REPRESENTATION ON DEMAND PATTERN

The didactical design pattern Representation on Demand gives advices to educators as well as learning tools developers to get the highest possible impact of learning. Most learning requires forms of representation such as written symbols or diagrams. For that reason, in lectures, lessons or even learning tools, multiple representations of the same (mathematical) objects should be provided. How this problem can be solved is described in the following didactical design pattern.

CHALLENGE / MOTIVATION (PROBLEM)	Contents, e.g. in mathematics, can be represented in many and different ways. For example, functions, in mathematical contexts, can be represented by algebraic terms, graphs, arrow-set-diagrams or a value table. E-learning tools only provide representations of the content, which fit to the context or the exercise. But learners are mainly on their own when working with e-learning tools, therefore the tool should offer more or all representations that the learner needs. With regard to theories of information reception, every learner should have the possibility to choose his/her best known and most comprehensible representation of the content according to their learning style. Only this way it is ensured that the learning potential is applicable.
FORCES	Offering all possible representations of a content to learners, e.g. in a lecture, costs a large amount of time. Usually, this time is not available or other contents have to be eliminated. Additionally, not all representations are needed. In many cases one or two representations are sufficient, and additional representations of the same content are boring for the learners. Multiple accesses to a topic are not always beneficial to all learners. In particular, the weak learners or students at the early learning stages get confused and overburdened by too many representations. Different descriptions of content require different approaches and perceptions, and therefore flexible handling of them. In computer tools displaying all of the representations of content would take too much working load and would take away “the view on the essential”. The GUI of a computer program would be too crowded and would move the main focus to the background. Users first have to get familiar with all of the representations, before he/she can start working or learning.
SOLUTION	In lectures one rarely has time to introduce more than two representations of a concept. One would, otherwise, loose too much time or having lectures only for showing representations. However, lecturers should provide two representation formats which are commonly used in the literature. Nonetheless, additional

	representations can be outsourced to bulletin boards or Learning Management Systems (LMS) and offered to students when needed. The learners can access these representations of content on demand. In computer learning scenarios or e-learning tools you can implement different kinds of representations. Beside the most commonly used presentation formats, the learner can enable additional representations when needed. In addition, the learner can try some of the yet "new" or unknown representations to obtain a new access to the content.
RATIONALE	When learning new content, several representations are often met. "Representations are any thing that stands for something else" (Schnotz, 1994). Manuals for technical products provide representations just as school content does. By connecting more representation formats of content the information level can increase (Kaput, 1989). Multiple representations can complement one another (Ainsworth, 1999) and contribute to a deeper comprehension. Different people need different representations concerning learning a new content (Vester, 1998; Bruner, 1968). As results of research on learning behaviours, learning contents should be presented with different representations for every type of learner. Hence, it can be ensured that the student can learn and assimilate the content optimally. The cognitive load-theory of Chandler and Sweller (1991) suggests that the working memory of our brain is limited. New information is first stored and processed in the working memory, and afterwards transferred to the long-term memory. Too many representations of content will overload the working memory and there is no space left for the learning content.
EXAMPLES	See following section "Actual Applications providing representation on demand".
RELATED PATTERN	Hint On Demand (Zimmermann, Herding, & Bescherer, 2013); Technology On Demand; Feedback On Demand (Bescherer & Spannagel, 2009)

Table 1: The representation on demand pattern

ACTUAL LEARNING TOOLS APPLYING REPRESENTATION ON DEMAND

Primarily, the didactical design pattern Representation on Demand was made to make learning for students easier by providing multiple representations of the contents of the lecture. For example, students can get additional materials through a learning management system (LMS) or in their weekly recitation groups, which contain a form of representation of a similar context to the lecture. Also, learning programs can be provided this way to show another perspective on the content.

Furthermore, learning tools can support the students learning if they follow the concepts of the didactical design pattern Representation on Demand. The learning tool has a chance to honour the learners' needs and demands by orientating on the patterns. Learning tools would feature more than one form of representations of the learning content but only make supplementary representations available if the students demand it.

The project SAiL-M (Semi-automatic Analysis of individual Learning-processes in Mathematics) has investigated applications on the didactical design patterns to the development and evaluation of the learning tools in teacher education in Germany from 2008 to 2012. The project has not only pointed out patterns (e.g. Bescherer & Spannagel, 2009; Bescherer, Spannagel, & Müller, 2008; Zimmermann, Herding, & Bescherer, 2013, also available on www.sail-m.de), it has applied them in an exemplary manner in the development of several learning tools and has evaluated the applicability of the patterns for them.

With the e-learning tool ColProof-M (Bescherer, Herding, Kortenkamp, Müller, & Zimmermann, 2012) students can verify simple geometric proofs, e.g. Thales' Theorem. The learner has to arrange a set of given logical propositions, and to state why each proposition is valid (fig. 2). On the one hand, the propositions are given in (mathematical) short notation as well as in plain texts. That means that those weak students can also work on the proof even if they do not feel confident with the mathematical notation. On the other hand, the students have the possibility to display the statement they have to prove via the dynamic geometry software (DGS) Cinderella (Richter-Gebert & Kortenkamp, 2012). Additionally, elements corresponding to the chosen proposition are highlighted. Every type of learner can select their way of representation when working on the task, using (short) mathematical notations or geometric manipulations.

SQUIGGLE-M (Fest, Hiob, & Hoffkamp, 2011) is a learning tool for the concepts and the properties of functions, also developed in the SAiL-M project. The software consists of several open learning laboratories. Each of them outlines a property of a function illustrated by one or more interactive forms of representations of the function. These representations also employ the DGS Cinderella. In some laboratories the function is represented as a term, a graph and a diagram (fig. 2), thus the learner can switch between these representations and get them connected or just choose the preferred representation.

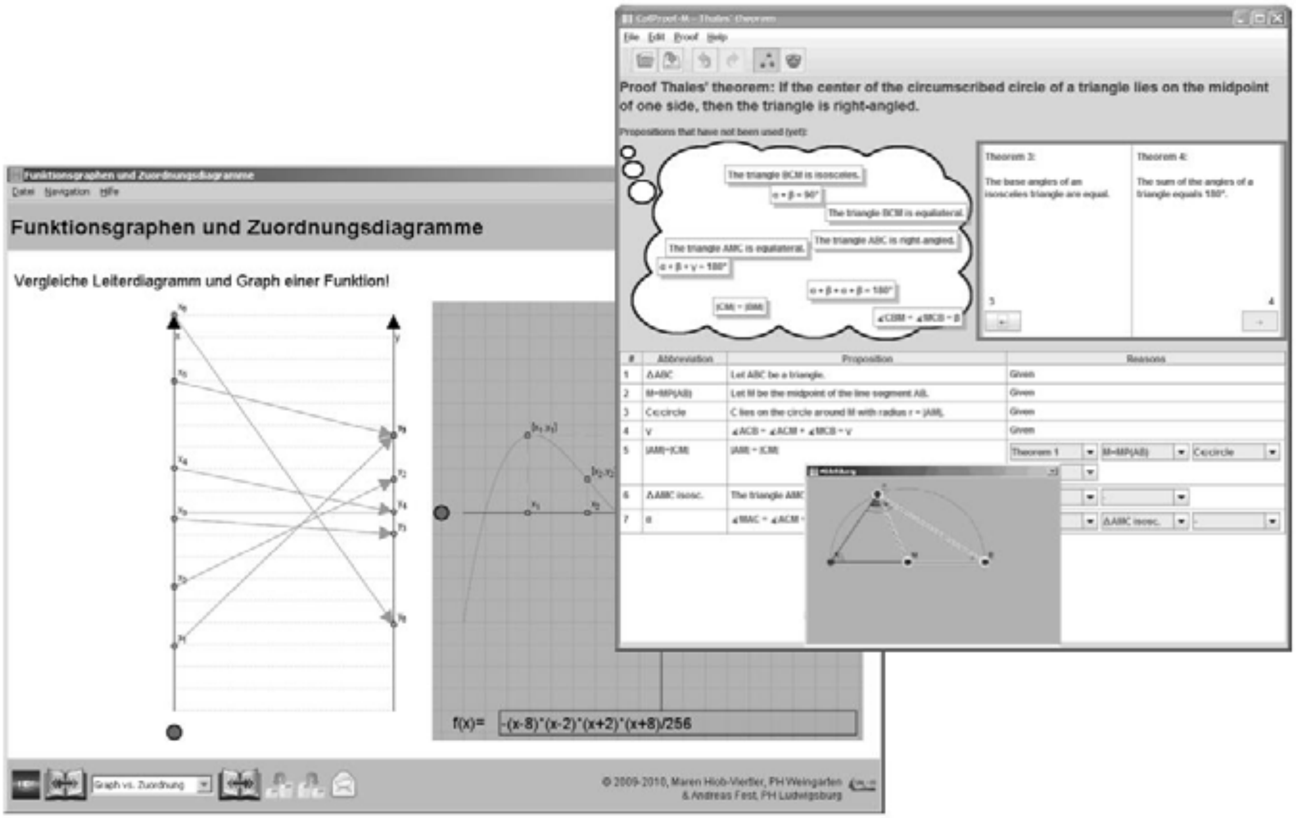

Figure 2: SQUIGGLE-M and ColProof-M with multiple representations

CONCLUSION

Several other didactical design patterns have been contributed within the SAiL-M project, most related to the usage of computer based learning tools. We refer to www.sail-m.de/sail-m/Patterns. Among others, Hints on Demand and Feedback on Demand are patterns that lie in the centre of the semi-automatic-analysis principles that have launched the project. They have been implemented in several learning tools which are run within a learning analytics architecture (Libbrecht et al., 2012). The Feedback on Demand pattern is implemented by a contact-teacher feature in the learning tools; feedback can be provided by the teacher because he can view the previous steps of the learning processes before responding.

This paper is a small contribution towards a greater visibility of the mathematics didactics to a broader public. The didactical design patterns we have outlined in this paper offer a simple and readable view of outcomes of the research in mathematics education. Their interpretation may support the software design process: the patterns may be embodied in user stories, their vocabulary may support the designation of software components or processes in as an interaction diagram. Examples of such user stories can be read at http://www.sail-m.de/sail-m/MoveIt-M_en but more research is needed into generalizing the application processes.

The contribution of this paper is, at the same time, an invitation for the mathematics education community to employ the format of didactical design patterns to describe mechanisms of the learning process as it appears to be appropriate to support the software construction process.

ACKNOWLEDGEMENTS

This work has been partially funded by the Ministry of Education and Research of Germany under the SAiL-M project and by the European Union's ICT-PSP under the project OpenDiscoverySpace http://www.opendiscoveryspace.eu/.

NOTES

1. In this article, we replace the name problem by the name challenge, because in pedagogy there is never one problem which occurs in the same way and which can be solved in the same way all the time. Instead, when challenges occur, they have to be solved with respect to the persons and context.

2. The German idea of didactics (Didaktik) means the science of learning and teaching of a specific subject i.e. didactics of mathematics or didactics of foreign languages. This definition is more specific than the general concept of pedagogy. In German, as in many of continental Europe's language, the negative connotation of didactics is absent.

REFERENCES

Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education, 33*, 131 – 152.

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language. Towns, buildings, construction.* New York: Oxford U.P.

Bergin, J. et al. (2012). *Pedagogical Patterns: Advice for Educators*. Joseph Bergin Software Tools.

Bescherer, C., Herding, D., Kortenkamp, U., Müller, W., & Zimmermann, M. (2012). E-Learning Tools with Intelligent Assessment and Feedback for Mathematics Study. In S. Graf, F. Lin, Kinshuk, & R. McGreal (eds.), *Intelligent and Adaptive Learning Systems: Technology Enhanced Support for Learners and Teachers* (pp. 151-163). Hershey, PA: IGI Global.

Bescherer, C. & Spannagel, C. (2009). Design Patterns for the Use of Technology in Introductory Mathematics Tutorials. In A. Tatnall & A. Jones (eds.), *Education and Technology for a Better World* (pp. 427-435). Berlin: Springer.

Bescherer, C., Spannagel, C. & Müller, W. (2008). Activating Students in Introductory Mathematics Tutorials. *Proceedings of the EuroPLOP 2008*.

Bruner, J. (1968). *Processes of cognitive growth: Infancy*. Worcester, MA: Clark University Press.

Chandler, P. & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction, 8*(4), 293 – 332.

Fest, A., Hiob-Viertler M., Hoffkamp, A. (2011). An Interactive Learning Activity for the Formation of the Concept of Function based on Representational Transfer. *The Electronic Journal of Mathematics & Technology, 2*(5).

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns. Elements of Reusable Object-Oriented Software.* Addison Wesley, Reading, MA.

Kaput, J. J. (1989). Linking representations in the symbol systems of algebra, In S. Wagner & C. Kieran (Eds.), *Research issues in the learning and teaching of algebra* (pp. 167 – 194). Hillsdale, NJ: Lawrence Erlbaum Associates.

Libbrecht, P., Rebholz, S., Herding, D., Müller, W., & Tscheulin, F. (2012): Understanding the Learners' Actions when using Mathematics Learning Tools. In J. Jeuring, J. Campbell, J. Carette, G. Dos Reis, P. Sojka, M. Wenzel, & V. Sorge (Eds.) *Intelligent Computer Mathematics, Proceedings, LNCS Vol. 7362*.

Niegemann, H. M. & Niegemann, L. (2008). Didaktische Entwurfsmuster: Idee und Qualitätsanforderungen. In P. Loos, V. Zimmermann, & P. Chikova (Eds.), *Prozessorientiertes Authoring Management: Methoden, Werkzeuge und Anwendungsbeispiele*. Berlin: Nomos

Noss, R. (2009), *Visions and challenges for technology enhanced learning research*. Invited talk at Journées Scientifiques EducTice, May 14th, 2009. Available from http://eductice.ens-lyon.fr/EducTice/ressources/journees-scientifiques/journees-eductice-2009/JE_EducTice/.

Richter-Gebert, J. & Kortenkamp, U. (2012). *The Cinderella.2 Manual*. Berlin, Heidelberg: Springer.

Schnotz, W. (1994). *Aufbau von Wissensstrukturen. Untersuchungen zur Kohärenzbildung bei Wissenserwerb mit Texten*. Weinheim: Psychologie Verlags Union

Vester, F. (1998). *Denken, Lernen, Vergessen*. München: dtv.

Vogel, R. & Wippermann, S. (2004), Communicating didactic knowledge in university education. Proceedings from *ED-MEDIA 2004 : World Conference on Educational Multimedia, Hypermedia & Telecommunications*, June 21 - 26, 2004, Lugano, Switzerland, (pp. 3231-3235). Norfolk, VA: AACE.

Zimmermann, M., Herding, D., & Bescherer, C. (2013). Pattern: Hint on Demand. In Y. Mor et al. (eds.), *Practical Design Patterns for Teaching and Learning with Technology.* Sense Publishers.