Flipped Classroom Patterns - Using Student Solutions

CHRISTIAN KÖPPE, RALPH NIELS, ROBERT HOLWERDA, LARS TIJSMA, NIEK VAN DIEPEN, KOEN VAN TURNHOUT, RENÉ BAKKER
HAN University of Applied Sciences, Arnhem/Nijmegen, Netherlands

The character of in-class meetings in a flipped classroom setting is certainly different from traditional lectures. Teachers in the flipped classroom tend to focus on giving feedback to the students, helping them with correcting misconceptions and providing support for deeper learning. Often, teachers in the flipped classroom use student solutions as a starting point, which is also described in the pattern USE STUDENT SOLUTIONS.
This paper proposes five patterns that help with using the student solutions in an effective way, by focussing on the aspects of motivation and respect.

Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science Education —*Computer science education*

General Terms: Design, Languages, Education

Additional Key Words and Phrases: Educational Patterns, Design Patterns

ACM Reference Format:

Köppe, C. et al. 2015. Flipped Classroom Patterns - Using Student Solutions – L(July 2015), 14 pages.

PLEASE NOTE: this is the version presented in a writer's workshop at the PLoP'15 conference.

1. INTRODUCTION

The flipped classroom approach to teaching was made popular by Bergmann and Sams with the release of their book *Flip Your Classroom* [Bergmann and Sams 2012]. It is characterized by moving instruction and introduction of learning materials outside of the class. The students acquire content on their own time by watching explanatory videos, reading supplementary material, and making assignments. The in-class meetings, or lectures, are changed too. The students are not listening to lectures, but will receive feedback on their home assignments, work (collaboratively) on deepening assignments, and hereby have the teacher available for questions and assistance in further inquiry. The lecture is therefore used for deepening the understanding of the content and better supporting the students with learning.

One of the biggest changes in a Flipped Classroom course compared to traditional teaching are therefore the activities of the students outside of the classroom and the activities of both students and teachers during the in-class meetings. Outside class, students are acquiring new knowledge through watching screencasts, readings, and by making exercises. The student solutions of these exercises form a valuable information source for the teacher, as she can use them to check whether the students have acquired new knowledge, how they have applied it, and where misconceptions are still present. Furthermore, these solutions are made by the students, so using them for giving feedback might make it easier for the students to understand that feedback, and to relate it to their own experience. We therefore consider it good practice to USE STUDENT SOLUTIONS during the in-class meetings for giving feedback, correcting misconceptions, discussing alternative solutions or pointing to possibilities for improvement.

authors addresses: christian.koppe,ralph.niels,robert.holwerda,l.tijsma,niek.vandiepen,koen.vanturnhout,rene.bakker}@han.nl

2. FOCUS OF THIS WORK

In previous work we introduced three patterns that generally address the usage of student solutions for the in-class meetings of Flipped Classrooms: USE STUDENT SOLUTIONS, EVERY STUDENT SOLUTION COUNTS, and COMPARE SOLUTIONS [Köppe et al. 2015a]. Summaries of these patterns are given in table I.

Pattern Name	SUMMARY
USE STUDENT SOLUTIONS	Use the work that students have handed in as examples in class.
EVERY STUDENT SOLUTION COUNTS	Make sure that each student in the group—or most of them—will see his or her work being discussed every once in a while so that they see the relevance of it.
COMPARE SOLUTIONS	Show and discuss multiple solutions so that students learn to recognize the strengths and weaknesses of various approaches, including their own.

Table I. : proposed patterns for design and execution of in-class meetings

In this work we introduce five more patterns that are intended as support for USE STUDENT SOLUTIONS [Köppe et al. 2015a] and that are also part of the SUITABLE CONTENT SELECTION [Köppe and Schalken-Pinkster 2013]. An overview of them is given in table II.

Pattern Name	SUMMARY
STUDENT CONTRIBUTION ESTEEM	Thank students when you use their solution in class as example or as trigger for discussions.
SOLUTION VARIETY	Use multiple student solutions that differ in approach or contain different solution directions for discussing the variety of important aspects of the concepts to be learned.
ANONYMIZE SOLUTIONS	If you use bad student solutions during in-class meetings for discussing their shortcomings, then initially anonymize them, so that the students won't be embarrassed and the focus is on the solution only.
BIRD'S EYE SUMMARY	Give a general overview of the strengths and weaknesses you identified in the students' solutions.
GENERALIZED FEEDBACK	Generalize issues of individual students when giving feedback on them so that the whole group can learn from them.

Table II. : proposed patterns as support for USE STUDENT SOLUTIONS [Köppe et al. 2015a]

3. PATTERN MINING SOURCE

In order to improve programming skills of students in computing at HAN University of Applied Sciences, we have successfully flipped two courses of 300 students each year since 2012-2013 [van Diepen et al. 2015]. This was based on two previous years of experience in a 25 student course. Theory and instruction in these courses is now studied at home by our students and they submit homework in advance of each lecture. During lectures, feedback is given on assignments, common mistakes are discussed, additional explanation is provided and homework for the next lecture is introduced.

We held two pattern mining workshops, where we collected user stories and best practices together with groups of lecturers who have given a Flipped Classroom course at least once. These stories and practices formed the base for a collection of pattern candidates (or proto-patterns). We selected five of the pattern candidates based on their relevance for supporting the application of the USE STUDENT SOLUTIONS pattern. These were further elaborated and all relevant pattern parts—context, problem, forces, solution, implementation details, and consequences—identified in an iterative approach with various rounds of writing and reviewing. The results of this writing process are described in this work. The complete approach and all results are described in more detail in [Köppe et al. 2015b].

4. THE PATTERNS

4.1 Pattern format

The patterns use an adapted version of the Alexandrian pattern format, as described in [Alexander et al. 1977]. The first part of each pattern is a short description of the context, followed by three diamonds. In the second part, the problem (in bold) and the forces are described, followed by another three diamonds. The third part offers the core of the solution (again in bold), the solution in more detail, the positive and negative consequences of the pattern application — which are part of the resulting context — and a discussion of possible implementations. This is followed by examples of the pattern implementation, shown in *italics*.

PATTERN: STUDENT CONTRIBUTION ESTEEM

Students have prepared solutions to exercises, and you want to USE STUDENT SOLUTIONS to organize an active classroom discussion. This discussion might involve that the solution owner, the student, talks about the reasoning behind his or her solution and that other students ask relevant questions and respond substantively.

❖ ❖ ❖

Discussing the work of a student is often experienced as personal critique and embarrassing for him or her, especially if done in front of the whole class.

Showing a student's solution can trigger all kinds of reactions that do not contribute to the learning process. Students might try to figure out who owns the solution under discussion, instead of focusing on the subject, or they might be commenting on the solution in an unproductive way.

In general, no one likes being the focus of attention if negative issues are discussed (even if they are presented as points for improvement). In consequence, not so good students might stop with handing in their homework.

❖ ❖ ❖

Therefore: thank the student explicitly for contributing to the specific aspect(s) of the subject matter you want to discuss with his or her solution and therefore helping everybody with learning.

Before starting the discussion about the solutions, you can explicitly thank the student for his or her contribution. However saying thank every time might feel a bit awkward and isn't necessary. The important aspect of the pattern is that you specify the contribution and acknowledge the work and thinking process of the student.

Asking questions to the solution owner about the solution provides you with information about the reasoning, but can also be used to implicitly thank the student. Make sure to signal that you value every answer and give the student enough time to think.

After listening to the student, relate the answer to the relevant aspects of the subject matter. Even if some answers are wrong, part of the reasoning might be valid, or at least relevant. Make sure you point out the relevance and chain the answer to the discussion you want to evoke by asking follow up question to the class.

Creating and maintaining a safe learning environment that thrives on students saying a lot of things is important. However, especially in the beginning of a new course where most students do not know each other well, you should consider starting with ANONYMIZE SOLUTIONS first in order to emphasize that the focus is on the solution used for learning and not on the student who handed it in.

In the programming course Object Oriented Program Development students are regularly asked to draw a memory model of an executing program. Most students have a difficult time understanding the memory model and there is a wide variety of misconceptions. Below are three student solutions for the same problem:

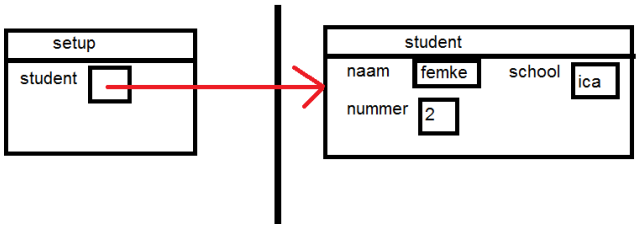

Fig. 1: Memory Model - version 1

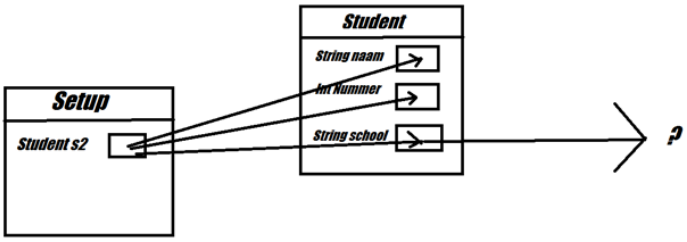

Fig. 2: Memory Model - version 2

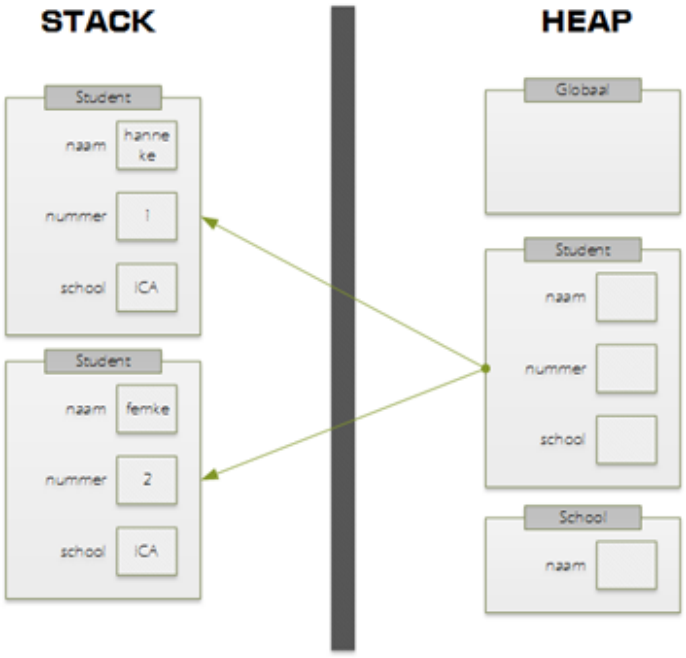

Fig. 3: Memory Model - version 3

In class the teacher thanks each student for sharing their thinking process, hereby helping everyone understanding the subject matter. To further reinforce self-esteem, correct elements are identified and the students are asked about their considerations when drawing the model. After that the solution is improved by using input from class and comparing it to other solutions.

PATTERN: SOLUTION VARIETY

You want to USE STUDENT SOLUTIONS for discussing your subject matter. You think about applying COMPARE SOLUTIONS as a good basis for the discussion.

❖ ❖ ❖

A single solution hardly ever contains everything that is worth focussing on and comparing a lot of solutions can become repetitive and take too much time.

Usually, students will have come up with very different solutions to the stated problem. Each solution has its own interesting aspects: things that have gone well can be interesting to highlight, but often you will want to focus on common mistakes too. If you pick random solutions you may miss important aspects.

To make sure that you highlight all interesting aspects in the solutions, you have to prepare a class by looking at all work handed in. This may take more time than is available.

It also takes too much time during the in-class meeting to discuss all solutions, especially in a bigger class. Furthermore, similar approaches will probably pop up in different solutions, which makes discussing them repetitive and probably not very useful.

❖ ❖ ❖

Therefore: discuss a selection of solutions with great variety.

It is better to make a selection of the solutions, so that every aspect that you want to discuss in class, is represented in at least one solution. By discussing the aspects of the subject matter using actual solutions, you will show that they are real (in the sense that they apparently show up in the work of students).

To make sure that you catch all interesting things about the solutions, you have to prepare well: you have to at least quickly scan each solution, but this may not be sufficient. You also need to be aware of the variety of approaches the students take when working through the preparation material.

Not always everything that you want to discuss about the subject matter can be found in the handed in solutions. In this case, you have to elaborate on the subject matter by other means, e.g. applying ADD VALUE BEYOND FEEDBACK.

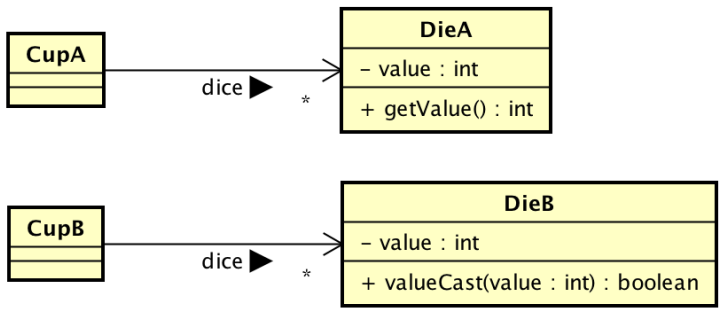

Fig. 4: Simplified class diagrams showing different approaches to solve the dice gambling game assignment.

In the first year course "Object Oriented Program Development" at HAN University of Applied Sciences, students learn to write programs in Java using basic principles from the field of object orientation. In one of the assigments, students

are to design and implement a simple dice gambling game, involving a number dice of dice and a cup that holds the dice. Students take various approaches: some decide to let the die communicate the number of pips on the rolled side, while others equip the die class with a boolean method that answers the question whether or not a certain value was cast. This difference in approach is a very interesting one to discuss responsibility of Java classes. The example is especially interesting to students because they brought up the difference themselves (although they were not aware of it before the teacher emphasized the difference). Figure 4 shows simplified class diagrams representing the two approaches.

PATTERN: ANONYMIZE SOLUTIONS

You have a class of students who don't know each other very well and you want to USE STUDENT SOLUTIONS in your in-class meetings as bad examples (e.g. for identifying and discussing the points of improvement).

❖ ❖ ❖

Students likely feel embarrassed in an unfamiliar group when their solution is used as a bad example and other students know who has handed in the work. The focus easily shifts from the solution to the student, which could result in students not handing in their preparations anymore.

Students who don't know each other at the beginning of a course have to find their place in the group. One of the things students use is comparing themselves with others with respect to their performance during the first time. Performing less well than other students might lower the position of a student, which is likely to be of negative influence on the motivation and self-efficacy.

❖ ❖ ❖

Therefore: anonymize the first solutions you use to emphasize that the focus is on the solutions and the learning opportunities they offer, not on the students who have handed them in.

This pattern is most relevant during the first in-class meetings, as it helps to create a safe environment where students are allowed to make mistakes without being laughed at. Students should get the feeling that learning from each other is more important than showing how good you are already.

In order to strengthen the emphasis on the solutions as learning opportunities instead of the solution's authors, you could also anonymize solutions that are used as good examples.

It requires usually very little time to just remove all information that could lead to the author. In case of e.g. source code files, one can make a copy of the work, rename it and remove all author-comments. Documents can be anonymized by removing the title page or copying the relevant parts to a new document.

However, the author usually still might be identifiable somehow (e.g. by purely asking everyone or by automatically added meta-data in a document). If that happens, then STUDENT CONTRIBUTION ESTEEM should be applied immediately to lower the negative effects and emphasize once more the focus on the solution and not the author.

Figure 5 shows an example of a handed-in solution (student's name replaced with author's name) and the anonymized version.

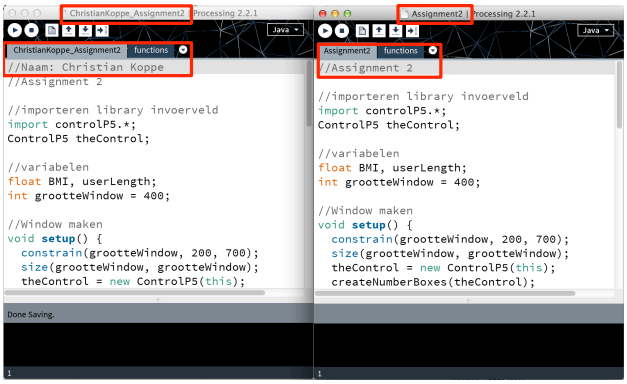

Fig. 5: The solution as handed-in by the student (left) and the anonymized version use in class (right).

PATTERN: BIRD'S EYE SUMMARY

You have just been discussing student's solutions to a particular homework assignment, applying USE STUDENT SOLUTIONS and possibly also ADD VALUE BEYOND FEEDBACK, COMPARE SOLUTIONS, or GENERALIZED FEEDBACK. Much has been shown and said in a short time, at different levels of abstraction.

❖ ❖ ❖

Students will find it difficult to extract the main points of the discussion from everything that has been said. Because the session in class will move swiftly to other topics, students may find it hard to retain the "take-home message" from this particular part of the session.

Basing a sizeable part of the classroom session on the discussion of a series of homework assignments creates a more segmented treatment of the subject matter than, for example, a well prepared, long(ish) lecture might be. There is less opportunity for a narrative arc that ends with a clear statement of the overall conclusions. And, if the teacher isn't careful about it, less attention is spent on integrating the different sub-subjects with one another, and with the overarching theme(s) of the session.

An interesting and effective discussion of a single homework assignment involves a lot of insights, examples, generalization and/or comparisons. Because time is, of course, limited, the information-density of the discussion tends to be somewhat higher in these flipped classroom sessions. All the while, the student is cognitively busy, integrating the insights presented with his/her previous mental model of the subject matter. More so, probably, than in traditional lectures, because he/she has already spent quite some mental energy on the subject matter while doing his preparation and assignments. In this situation, students may find it more difficult to distill the important statements or concepts than in a traditional lecture.

❖ ❖ ❖

Therefore: End each segment in the session (e.g. each discussion of a single homework assignment) with a very brief summary of the main points. Use this moment to connect the important point of the current segment with the points of previous segments.

Just a few sentences is usually enough for this purpose. Reiterating the main points, or explicitly restating the "take-home message", of this current topic of discussion, has some benefits to the students:

—It allows the student to perform a mental double check: "Did I understand this correctly?"

—It helps students who have difficulty in separating main points from side-issues.

—Knowing that a summary will follow the discussion, it may allow the student to spend more cognitive effort on integrating the new insights with pre-existing knowledge and earlier experiences.

—It signals the end of the current segment, adding to the perception of structure and progress in the session.

This moment of summary is also an excellent opportunity to integrate the subtopics of the different segments by referring briefly to them. This helps restore the student's overview of the relations between all the different concepts of the session. It also aids in the retention of earlier "take-home messages".

If possible, one can refer to subtopics that will appear in upcoming segments, providing a bit of a "preview" of the rest of the session. Referring to the topic of the immediately succeeding segment allows you to provide a comfortable segue into the next part of the session.

One session in the programming course "Scripting for Designers" at HAN University of Applied Sciences is dedicated to the topic of different data types in programming languages (such as text, yes/no values and different kinds of numbers). At one point in the homework assignments, students are asked to write a program that repeatedly adds the number

0.1 to itself, and prints the results. Students will observe that the first few results (0.1, 0.2, 0.3, 0.4, ...) are as expected, but after about eight rounds, the result acquires some unexpected numeric "residue": the program shows values like 0.800000000003405.

In the classroom discussion of this experiment, the teacher explains that this is because computers store numeric values in a binary notation (as opposed to the decimal notation that we humans are used to). The value 1/10 has, in binary notation, an infinite sequence of digits after the dot (similar to the way the value 1/3 has an infinite sequence of 3's after the dot in decimal notation). This forces the computer to round the number to fit in a typical memory cell. This rounding is very subtle, and only becomes apparent after about 8 additions in our experiment.

After this explanation (spiced with some more examples of unexpected behavior of numbers in computer programs), the discussion is terminated by stating the take-home message: In typical programming languages, numbers with decimals after the dot are slightly unreliable. Do not assume exactness of results.

This summary tells the students that they don't have to understand or remember any intricacies of binary notation that were demonstrated. It's the unreliability that matters.

Finally, the summary refers to an earlier discussion of the reasons why most programming languages require programmers to choose between two numeric types: whole numbers, and numbers with fractions. The current segment just added another reason: the type of whole numbers is reliable, whereas numbers with fractions aren't.

PATTERN: GENERALIZED FEEDBACK

You have just been discussing student's solutions to homework assignments (as described in USE STUDENT SOLUTIONS). Your intention was to give students a deeper understanding of the subject at hand, and to make them see how the principles involved are useful to them.

❖ ❖ ❖

Students may find it difficult to distill the main points or the important principles from the example problems and the student solutions under discussion.

The discussion of the subject matter and its important principles had been grounded in the concrete example problems that were posed in the homework assignments. Sometimes it's possible to provide several different homework problems that involve the same abstract knowledge or skills in different ways. Oftentimes, though, there isn't enough time for that.

Furthermore, the discussion has focused on solutions provided by a subset of the group. Especially when the homework assignment is interesting enough that it allows for several different kinds of solutions, quite a bit of the discussion will (and should) be focused on the merits and drawbacks of these different ways of applying (probably) the very same principles.

Many students prefer discussion of concrete issues above treatments of abstract principles. Although the abstract principle is much more useful to them, it's also more difficult to grasp and more likely to be considered boring by these students.

A challenge to the lecturer, then, is to enable the transition of the discussion (and student's understanding) from the concrete example problems and their concrete example solutions to a generalized understanding of the principles involved, and the main points that the lecturer wants to make.

❖ ❖ ❖

Therefore: Interleave the discussion of the concrete aspects of the examples with explicitly mentioning the general principles involved. Inject small examples of other situations where the same principle can manifest itself in a different way.

For many students, working from the concrete issues/examples up to a generalized understanding of abstract concepts is more likely to succeed than the reverse approach that works from abstract principles down to concrete examples. This pattern describes a full-circle approach: from a concrete problem (in the homework assignment) with its concrete solutions (under discussion in class), move to the more abstract level of general principles. Then use the additional (small) examples to base the discussion of the abstract level by providing a connection back into the concrete level (and to support the generalization). It is therefore related to SOLUTION BEFORE ABSTRACTION.

By explicitly calling out the general principles, you're helping students elevate their understanding, and empowering them to remember and apply those principles in new situations/problems.

Also, you're connecting the student's efforts (in doing the homework, and coming to class) more explicitly with the learning objectives of the course. This increases their perception of those efforts as valuable.

By adding small examples of other situations or problems that involve the same principle you're, in effect, "proving" to the students that the principles indeed do have a more general applicability. You are also demonstrating that seeing the abstraction is useful, because it simplifies the understanding of a wide variety of concrete issues.

By introducing some or all of the general principles in class, you're positioning their introduction *after* the student's have spent some time and mental energy on the problems or phenomena. This will make them more receptive to the general principles that otherwise would be seen as (even more) boring.

In the programming course "Scripting for Designers" at HAN University of Applied Sciences, one assignment involves writing a program for finding the longest piece of text in a list of texts. This assignment is paired with an earlier assignment

in which they had to analyze a given piece of programming code, to discover that it calculates the sum of a list of numbers. The solutions to these problems are only similar on a somewhat more abstract level. They both involve a “loop" (repeatedly executing the same instructions on different data), and the designation of a piece of computer memory to collect the partial result while the loop hasn’t finished yet. This is called an “accumulator". During the discussions in class, the general principle of an accumulator is distilled from the two examples, and made explicit. Some examples of other uses of accumulators are shown, from domains such as databases, game development and financial calculations.

In this case, the material that students studied at home, before working on these assignments, were videos that introduced features of programming languages for working with lists of data.

5. CONCLUSION

In this paper we presented five patterns that help with using student solutions during the in-class meetings of Flipped Classrooms. We will continue collecting and describing more patterns that help with the design of valuable in-class meetings and other aspects of Flipped Classrooms with the goal of developing a pattern language for flipping the classroom.

6. ACKNOWLEDGEMENTS

We thank our shepherd Mauricio Aniche for his suggestions and comments.

REFERENCES

ALEXANDER, C., ISHIKAWA, S., AND SILVERSTEIN, M. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press.

BERGMANN, J. AND SAMS, A. 2012. *Flip your classroom: Reach every student in every class every day*. ISTE.

KÖPPE, C., NIELS, R., HOLWERDA, R., TIJSMA, L., VAN DIEPEN, N., VAN TURNHOUT, K., AND BAKKER, R. 2015a. Flipped Classroom Patterns - Designing Valuable In-Class Meetings. In *Preprints of the 20th European Conference on Pattern Languages of Programs, EuroPLoP'15*. Irsee, Germany.

KÖPPE, C., NIELS, R., HOLWERDA, R., TIJSMA, L., VAN DIEPEN, N., VAN TURNHOUT, K., AND BAKKER, R. 2015b. Towards a Pattern Language for Flipping the Classroom in Computer Science Education. In *unpublished*.

KÖPPE, C., PORTIER, M., BAKKER, R., AND HOPPENBROUWERS, S. 2015. Lecture Design Patterns: More Interactivity Improvement Patterns. In *Preprints of the 22nd Pattern Languages of Programs conference, PLoP'15*. Pittsburgh, USA.

KÖPPE, C. AND SCHALKEN-PINKSTER, J. 2013. Lecture Design Patterns: Laying the Foundation. In *Proceedings of the 18th European Conference on Pattern Languages of Programs, EuroPLoP'13*. Irsee, Germany.

PEDAGOGICAL PATTERNS EDITORIAL BOARD. 2012. *Pedagogical Patterns: Advice for Educators*. Joseph Bergin Software Tools, New York, NY, USA.

VAN DIEPEN, N., CHUNG, R., HOLWERDA, R., TIJSMA, L., TREFFERS, J., VAN TURNHOUT, K., AND BAKKER, R. 2015. Large Scale Flipped Programming Courses. *INTED2015 Proceedings*, 1162–1168.

APPENDIX A

Overview of referenced patterns

Pattern Name	Summary
ADD VALUE BEYOND FEEDBACK [Köppe et al. 2015a]	Interweave feedback with added value moments.
COLLABORATIVE EDITING [Köppe et al. 2015]	Use a tool which offers access to the same content for you and the students and collaboratively edit this content with the tool.
COMPARE SOLUTIONS [Köppe et al. 2015a]	Show and discuss multiple solutions so that students learn to recognize the strengths and weaknesses of various approaches, including their own.
SOLUTION BEFORE ABSTRACTION [Pedagogical Patterns Editorial Board 2012]	Let the students first find solutions to specific concept-related problems, have them identify the common aspects of these solutions, and use these identified aspects to introduce the general, abstract concept.
USE STUDENT SOLUTIONS [Köppe et al. 2015a]	Use the work students have handed in as examples in class.

Table III. : Patterns that are relevant for Flipped Classrooms