

Lecture Design Patterns: Improving Interactivity

CHRISTIAN KÖPPE, HAN University of Applied Sciences, Arnhem, the Netherlands

JOOST SCHALKEN-PINKSTER, HU University of Applied Sciences, Utrecht, the Netherlands

Many universities still use lectures as a form of teaching. Although lectures are often described as ineffective or inefficient, they can nonetheless be a valuable teaching form. The perception of ineffectiveness and inefficiency of lectures stems from the expectancy that lectures can only be a one-way form of communication from teacher to students. Since it has been demonstrated over and over again that one-way communication is an ineffective form of teaching, a key aspect of good lectures is interactivity with the students.

Some pedagogical patterns covering the interactive aspect have already been described in the pattern literature. In this paper we describe five additional patterns with the intent to increase the body of knowledge on making lectures more interactive and therefore more valuable. These patterns are: REMOTE HAND, QUESTION BOOMERANG, STUDENT MINERS, QUESTION PARKING SPACE, and COLLABORATIVE SUMMARY.

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

General Terms: Education, Lectures, Interactivity

Additional Key Words and Phrases: Educational Patterns, Active Students

ACM Reference Format:

Köppe, C., Schalken-Pinkster, J.J.P. 2013. Lecture Design Patterns: Improving Interactivity – L(October 2013), 15 pages.

PLEASE NOTE: this is the author's version of the paper, published in the Proceedings of the 20th Conference on Pattern Languages of Programs, PLoP'13.

1. INTRODUCTION

Few things have such a profound overall influence on a person's career and therefore a person's life as one's education. Hopefully everyone has had the experience of effective and inspiring education at least once in their life. In such cases, the teacher's stories were inspiring, the activities were thought-provoking and learning was fun and a source of energy instead of mind-numbing and energy-draining. What are the differences between good, effective and inspiring education on the one hand and mind-numbing, energy-draining education on the other hand? In this paper we discuss aspects of good, effective and inspiring education, namely that of interactive lectures in the context of Computer Science education. We chose educational design patterns as the format for describing these aspects.

Quite a few of such educational patterns have already been published. Some of these patterns also cover aspects of the design of good lectures. However, many of these patterns were originally not specifically intended for lectures or are very specific, e.g. regarding the lecture content.

In another paper [Köppe 2013b] we started to collect existing patterns for lectures and identified some categories for these patterns. These categories can be used for supporting pattern selection and offering a structure. We also described basic patterns underlying lecture design in earlier work [Köppe and Schalken-Pinkster 2013]. Two of these basic patterns—REGULAR ATTENTION RECUPERATION and LECTURE STRUCTURING—already lead towards including interactivity consciously in lecture design. But these basic patterns are described at a very high level of abstraction and

Author's address: christian.koppe@han.nl, joost.schalken@hu.nl

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in a writers' workshop at the 20th Conference on Pattern Languages of Programs (PloP). PLoP'13, July 23-26, Monticello, Illinois, USA. Copyright 2013 is held by the author(s). HILLSIDE 978-1-941652-00-8

therefore need patterns that are more specific and more concrete in order to implement these basic patterns. This paper describes some more patterns that help to make lectures more interactive and the paper provides an overview of existing patterns that serve the same goal.

In this paper we propose the following five patterns: REMOTE HAND, QUESTION BOOMERANG, STUDENT MINERS, QUESTION PARKING SPACE, and COLLABORATIVE SUMMARY. The patterns are more specific variations of the generic ACTIVE STUDENT pattern, which has been documented by the Pedagogical Patterns Project [Pedagogical Patterns Editorial Board 2012] and is stated as follows:

ACTIVE STUDENT

You want to maximize student learning. Passive students don't learn much. If students listen to explanations, without themselves becoming engaged, what is learned is unlikely to go into long term memory. The deep consequences of a theory are unlikely to be obvious to one who reads about, or hears about the theory. The unexpected difficulties inherent in using the theory or applying the ideas are not likely to be apparent until you actually do use the theory. Readings, lectures, and multi-media demonstrations, unless interactive, leave students passive.

Therefore: keep the students active. They should be active in class, either with questions or with exercises. They should be active out of class. Reading is often insufficiently active. Short readings should be followed by activities that reinforce what has been learned in the reading. The same is true of information given verbally or even visually through multi-media visualizations. If the students don't actively engage the material, they won't retain it. They need to write and they need to "do."

The patterns QUESTION PARKING SPACE and QUESTION BOOMERANG have to do with stimulating student involvement and with actively assessing their own learning achievements by honoring questions (even if they do not fit in the current lecture story-line) and by involving students in the answering of questions. The patterns STUDENT MINERS and COLLABORATIVE SUMMARY have to do with the active involvement of students in creating the theoretical starting point of the lecture and creating closure of the lecture. The pattern REMOTE HAND involves students in the use of tools during the lecture.

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*.

All related patterns are highlighted in SMALL CAPS. However, we moved the citations of these related patterns to the end of this paper, as these citations might hinder a smooth reading and thereby distract the reader.

REMOTE HAND

You want to demonstrate a tool during a lecture, where a tool can be any piece of software: graphics application, a programming language, a command line interface, etc. The students will have to use this tool afterwards or execute tasks similar to the ones you are going to demonstrate.



Presenting the tool all by yourself places the students in a consuming role. Yet it takes too much time and causes disorder to let different students try out the tool in front of the class so that everyone can see or understand what's going on.

One common way of learning tools is by *trial and error*. But during teacher demonstrations often only the correct way of using a tool is shown, which is different from the naturalistic trial and error way of learning. When not showing errors during a demonstration, students do not learn to diagnose and fix common mistakes while using the tool.

Having all students trying out the tool during the lecture makes it hard if not impossible to help all individual students with learning the tool. Most of them will be *missing feedback* from you while trying the tool out.



Therefore: let the students tell you what to do and then execute it so that everyone can see it. Combine the tool's feedback with your own.

All tools give feedback in some way. This can be either by doing exactly what you want and presenting a correct result, but also can consist of error messages or unexpected results. All of this tool feedback can be used perfectly for didactical purposes and the tool feedback can be combined with your own FEEDBACK on the students' suggestions, minimizing the chance of missing feedback.

Sometimes students ask things about the tool that even you as teacher do not know. In that case you can combine this pattern with QUESTION BOOMERANG and ask the students what they think how they expect it to work. Then do what they tell you to and examine collaboratively the feedback and results, therefore applying trial and error as whole group.

To optimize the learning experience, the teacher should summarize the correct way of tool usage after a trial and error demonstration, to ensure the students maintain an overview of the process of using the tool.

This solution might help to immediately address problems that students otherwise would have experienced later when working with the tool.

There is a chance that more experienced students will take the lead in such an activity and that other students start to see it as 'their game'. This might discourage these other students. It might also be that some of the suggested actions are too high-level or do not support the learning objectives of the lecture. In the latter case it might be better to use the QUESTION PARKING SPACE and answer these questions after the lecture in direct contact with the students.

REMOTE HAND can be combined with EXPOSE THE PROCESS, as the students might explore the tool by themselves and discuss possible steps in a process while you're executing them, visible for all.

Making this pattern work requires some preparations. You need to CHECK PREREQUISITES. A helpful list is as follows:

- The tool needs to run smoothly on the machine you're using for the presentation.
- The projection area should be big enough and all elements on the screen should be clearly readable (or recognizable) for all students, including the ones in the last rows of the lecture room.
- You should make some preparations for guiding the students in this delivery form. Leaving the process completely open for the students might confuse them or increases the chance that the actions the students ask you to do are not going to help to achieve the learning objectives of the lecture. These preparations might include CAREFULLY CRAFTED QUESTIONS or a step-by-step guide towards a certain result (including enough room for variety).

—As it is sometimes hard to know upfront how much time a REMOTE HAND exercise will require (e.g. due to digressions), it is good to include some BUFFERS (as part of LECTURE STRUCTURING) when you select remote hand as delivery form.

Sometimes you want to use REMOTE HAND for a text-based artifact. If that is the case and all students in the lecture have access to the internet — either via their own laptop or when the lecture is held in a computer lab — then online collaborative editing tools can be used. In that case make the students type their answer into the shared document and copy this document to the text editor (like a programming IDE). Examples of suitable online tools are *CollabEdit*¹ and *Google Docs*². These tools for online text editing can allow anonymous editing (like CollabEdit) or not (like Google Docs), remember that using an anonymous editing tool in combination with a less mature group of students can lead to unwanted edits (i.e. digital vandalism).

REMOTE HAND is a part of MAKE IT THEIR PROBLEM, but more broadly applicable. It is also a good addition to both TEST TUBE and TRY IT YOURSELF, as it offers a way for students probing the machine or trying something out without the need for a laboratory. Both SHOW IT RUNNING and SHOW PROGRAMMING can benefit from REMOTE HAND too, as it adds some interactivity to them and therefore more actively involves the students.

In a lecture on design patterns we applied DISCOVER YOUR OWN PATTERN by giving the students a design problem and some initial questions. The design problem was that a file-system was needed whereby easily can be determined the number of files in a directory including all sub-directories. The intention was to work towards the COMPOSITE pattern [Gamma et al. 1994]. The lecturer opened a UML-editor and started with the first CAREFULLY CRAFTED QUESTION: "Which general elements do we need?". Based on the answers of the students he modeled the classes they mentioned. Then the teacher asked how the classes (file and directory) are associated, thereby immediately modeling the answers in the tool visible to everyone. If there were alternatives mentioned, then these were modeled too, so that they were visible to all students, which made discussing them much easier. Then the teacher went on with asking for specific problems of the current state of the design and how they could be solved, continuing until the structure of the COMPOSITE pattern has been found by the students. In that case REMOTE HAND was a good combination with STUDENT MINERS.

REMOTE HAND was also applied in a course called "Structured Program Development", which is an introductory programming course. The course was designed using the flipped classroom approach, so the students had to work on assignments during the lecture and the results were discussed with the whole group. To make this discussion easier, the students had to copy their code to a collabedit-webpage. This way it was visible to everyone, both on the beamer projection and in the browser on their own machines. During discussion some students also made suggestions for improvement of the code, which was realized by the lecturer using REMOTE HAND.

¹<http://collabedit.com>

²<http://docs.google.com>

QUESTION BOOMERANG

A question is asked during a lecture.



If all answers are given by you, the students start to rely on you only. They don't recognize that they sometimes already know the answer to the question and stay more passive.

A teacher who provides all the answers herself increases the chance that the students see the *teacher as the only source of information*. All new questions that arise will be asked to her, and only her, instead of stimulating the students to look for other sources as well.

Answering all questions by yourself also *disengages the students* who know the answer and encourages them to withdraw.

From the students' perspective, directly asking questions that come up, instead of thinking first about the answer for themselves, increases a certain *thinking laziness*, which in consequence leads to less active students.



Therefore: Send a question back to the students like a boomerang. Let them try to answer it themselves as group or to think of possible answers. Do this repeatedly until they found the answer themselves or the question requires the introduction of new material.

You should always HONOR QUESTIONS, because when students are asking questions it is most often an indication that they are actively engaged in the lecture and with the content. However, in many cases some or most students already know the answer or can deduce it from their existing knowledge. Throwing the question back to the students helps them to activate that knowledge or to think about new ways of applying their existing knowledge. This is an implicit application of LINKING OLD TO NEW and EXPAND THE KNOWN WORLD.

Encouraging the students to answer helps to develop the class into a community of learners who are learning together, instead of merely a group of individuals learning from another individual (the teacher). It also shows them that they as a class already possess some knowledge which they can share. The class no longer has to rely on the teacher as the sole source of information anymore and might in some cases even lead to short-term INVISIBLE TEACHERS.

The simplest way of implementing the solution of this pattern is to ask "What do you think yourself?" . This mainly applies if you want the student who asked the question to answer it by herself. If you want the whole group to think about the answer, then one possible way of bringing the question back to the group is "Does someone else know the answer?" or — in a somewhat less strong way — "Who has an idea on how to approach this question?". You should stimulate the students to provide an answer in their OWN WORDS instead of just repeating definitions or terms, so that you can also judge if they really understood what they are talking about.

When applying this pattern it helps to repeat the question if you're not sure that all students heard and understood the question. A simple application would be stating "So the question (asked by x) is . . .". Mentioning the student again who has asked the question is not necessary, but might sometimes be a good way to HONOR QUESTIONS and make it more personal. Sometimes the question asked is somewhat vague. If you think you've got the intention of the question, then try to rephrase it first and let the student confirm that this indeed was his or her question. If the question doesn't make sense at all, ask the student to rephrase it or to give a possible example of what he or she is thinking about.

Following is a summary of common strategies for applying this pattern:

- Simply return the question by asking: "What do you (the other students) think?". This is the most common strategy.
- In case of questions that can simply be answered by "yes" or "no", but where this answer is related to other important implications, then it is helpful to send the question back in a different way, namely by asking: "What would happen if you do x?" or "What would the consequences be if y would be possible to do?". This kind of question triggers a deeper thought process than just having a "yes" or "no" answer.

Even though one might have taken into account that applying this pattern might take more time (as it is sometimes hard to predict how the students will answer and where some follow-up discussions will lead), it is generally a good idea to include sufficient BUFFERS in the lecture. If you have a plan for your lecture and the discussion is still ongoing, but you feel that the discussion is valuable, then think about to LET THE PLAN GO. If the question requires an answer that you cannot give at this moment (for whatever reason), then make use of a QUESTION PARKING SPACE to ensure that it will not be forgotten and answered later.

Over-applying this pattern can have some negative consequences. If all questions are sent back to the students they might start to miss the value of asking questions (if they always have to answer them by themselves). So one needs to find a good mix between QUESTION BOOMERANGS and answering questions directly. The main criterion for applying QUESTION BOOMERANG is probably if a discussion about possible answers is pedagogically useful. On the other hand, if the question is very specific, then it's probably better to answer it shortly and to continue with the lecture.

In the case of really simple (closed) questions, applying QUESTION BOOMERANG might be an overkill. Such simple questions often just should be answered directly.

In a lecture on Introductory Programming the concept of loops as programming construct was introduced. During the discussion on the for-loop, one student asked what happens if one uses the decrement operator “-” for the loop-variable instead of the usual increment operator “++”. The question was sent back to the whole class, they were asked: “What do you think will happen when we change the operator? Who has an idea?”. This way the students were triggered to think about the consequences of using the (in that case) wrong operator (leading to an infinite loop, as the guarding-condition will never become false) and encouraged to discuss the possible results.

STUDENT MINERS

Also known as: Collaborative Knowledge Construction.

You want to introduce a new concept, which is related in part to concepts the students already know.



Just presenting a new concept makes it hard for students to relate this new knowledge to their existing knowledge and keeps them in an undesired passive role.

Introducing a new concept in isolation is likely to be less effective because of the *missing link to existing knowledge*. It costs more time afterwards to explain the links between existing knowledge and the new concept.

Even when you make the link more explicit by presenting it in your lecture, it still does not actively reactivate this existing knowledge in the students. The students stay *knowledge consumers*. There is a difference if the students are told what they already (should) know or if they have to reactivate this knowledge by themselves.



Therefore: Introduce the concept through questions that are related to existing knowledge and lead towards the new concept; don't present the concept yourself directly. Let multiple students provide a variety of answers to these questions and lead the group through follow-up questions towards the new concept. Mine the new concept from all answers together with all students.

This pattern is similar to BUILD AND MAINTAIN CONFIDENCE and REFLECTION, which both focus more on having the students find solutions to a given problem by themselves. But the core idea is the same: do not present something by yourself that the students are about to learn, but let them find out about it (mainly) by themselves, based on their own knowledge and experiences.

STUDENT MINERS can best be used if a student already has some relevant prior knowledge, as the pattern is less effective when introducing an new field of theory to students. Therefore this pattern is best used with students that are not in their first year of studies.

Applying this pattern might increase the students' perception that they have discovered the concept by themselves (as they did EXPAND THE KNOWN WORLD of their own knowledge), which is a good learning motivator.

PROBLEM ORIENTATION could be a good starting point for the STUDENT MINERS. CAREFULLY CRAFTED QUESTIONS trigger the students to think about a topic in different ways, the variety of answers could be used to approach a new topic starting from a students' perspective (as they gave the answers) and to discuss different views. In order to trigger the thought process more deeply, these should be OPEN ENDED QUESTIONS. Make sure to use a PREGNANT PAUSE so that the students have enough time to think about their answers. UNINTERRUPTED LISTENING is necessary in order to really hear all students' thoughts.

The implementation of the pattern solution often follows a certain flow: (1) the lecturer should start with questions for reactivating existing knowledge, (2) then a new problem can be introduced where this existing knowledge forms the basis or starting point for finding the solution of the problem, and (3) the following questions should help the students to make the connection between existing knowledge and the new problem by combining or varying parts of their existing knowledge and exploring possibilities of filling in the missing parts.

During application of this pattern, the lecturer might get a lot of answers. Even if the first answers already include the correct one, ask for more answers. Start a discussion about the differences between these answers and look for common parts in the answers. Let the students identify these themselves, as this will making the learning effect stronger.

When having the students mine their own knowledge and collaboratively working towards the new concept, it is a good idea to use NAME IS LAST. That way you can ensure that they really focus on the concept itself and are not distracted by the name and some (possibly) wrong associations the students have with it.

The hard thing about this pattern is that you as teacher do not know in advance what the students' answers will be. You should therefore be able to react in a forward-leading way to a variety of answers. This is not easy and some teachers might find it too difficult or might feel uncomfortable. However, you always have the option to stop the STUDENT MINERS and just present the concept by yourself, which would result in a situation as if this pattern solution hasn't been applied.

A more concrete variation of STUDENT MINERS is the pattern DISCOVER YOUR OWN PATTERN, which is part of a pattern language for teaching design patterns. Here the existing knowledge is the one about (object-oriented) software design principles like encapsulation, loose coupling, etc. By making use of this knowledge and introducing a problem — like "one needs to be able to easily add or change the way the bonuses of employees are calculated" — one can easily lead the discussion towards the structure of the STRATEGY pattern [Gamma et al. 1994].

There are some similarities between STUDENT MINERS and MAKE IT THEIR PROBLEM, but contrary to MAKE IT THEIR PROBLEM the STUDENT MINERS do not have to agree upfront on a solution to the problem. Instead the focus of the STUDENT MINERS is to collaboratively find the solution and to discuss alternatives, hereby identifying some underlying concepts on their own.

STUDENT MINERS is a more conscious application of LINE OF REASONING, whereby you are not reacting on an unexpected answer of a student to one of your questions, but where you are asking the questions in such a way that you might get different and also unexpected answers.

In a course on Object-Oriented Analysis and Design I wanted to introduce the concept of generalization/specialization on a conceptual level (as part of domain modeling). I had already used an example for introducing other parts of domain models —like conceptual classes, attributes, and associations— which was a simple room reservation system at a university. We discussed collaboratively that both students and teachers should be able to make room reservations, e.g. working on a project or for having a team meeting. So we had on the whiteboard the conceptual classes reservation, student, and teacher. There were associations between reservation and both student and teacher. I asked them what the association exists between reservation and teacher if a student has reserved the room and vice versa. We discovered that this might be a problematic flaw in the design and the students started to discuss alternatives, like adding a flag that signals if the student or the teacher has reserved a room. At some point in the discussion one student mentioned anyway it's always one person who has made the reservation, and we should link the reservation to this person. I asked what the difference is between the concepts person, student, and teacher, and another student answered that both students and teachers are persons. At this moment I gave the concept the students just had discovered a name — generalization. We then continued this discussion (The question "can we ever have a person in our little system who is neither student nor teacher?" lead to the introduction of abstract conceptual classes...).

QUESTION PARKING SPACE

You're presenting content in a lecture and made clear that you HONOR QUESTIONS.



Students often have questions that do not directly relate to the content or require a longer or very specific answer. Answering such a question immediately might disturb the flow of the lecture, but not answering them at all might result in an unsatisfied student.

Not answering the questions asked is an indirect way of *not honoring questions*. This increases the chance of discouraging the students to go on with answering questions.

Answering all questions immediately might require more time than you have for the overall lecture, even with sufficient BUFFERS. As result there is a chance that you are *running out of time* at the end and won't be able to finish the lecture as planned.

If a question is asked where the answer is mainly of *individual interest* for the student asking the question, but not the whole group, then answering this question might bore most of the group and lead their attention towards non-lecture activities.



Therefore: If a question comes up during a lecture that cannot be answered directly —either because you don't know the answer or because you don't have the time or possibilities to answer it— put it in a parking space and provide an answer at a more suitable time or in a more suitable way.

The best is to write the question down. If you can answer it later in the lecture, then it could be sufficient to remember the question yourself (make sure that you really remember it!) or to ask the student to remember it for later. However, the most important aspect is that the question will be remembered and answered at a later moment.

By making sure that all questions are answered, even though not immediately, you still show that you HONOR QUESTIONS. The question parking space can also be used if you don't know the answer. In that case you should be honest and show the students that NOBODY IS PERFECT or use QUESTION BOOMERANG as alternative.

If you decide to write the questions down, then of course you need some place for this like a whiteboard or a flipchart. However, this also means that this place might become occupied by these questions and is not available for other activities during the lecture. It is therefore recommendable to use a place that is not required during the lecture. Examples are a special sheet on the flipchart you can flip to if necessary or a simple text editor you keep open in minimized mode during the lecture.

Some answers are only of interest to the student who has asked the question, e.g. if it's related to his or her work or to a personal project. Answering such questions during the lecture exposes the other students to information that is probably irrelevant for them. In that case it is better to give the answer during BREAKS —if applied— or right after the lecture.

Be aware that writing down questions takes time. The longer it takes, the higher the chance that the students draw their attention to other things which means that you have to put some extra effort into getting them back on track. You should therefore try to write down the question as fast and non-interruptive as possible. It hereby is helpful to either find one or two remarkable words that help you (and/or the students) to remember the question (e.g. the words "nested loops" in the example below). An alternative is to use a symbol (e.g. a metaphor) or a piece of an example that represents the question (e.g. the lollipop-notation for UML interfaces).

Remembering the questions without writing them down has the risk that you forget some of them. This might give the students the picture that their questions are factually not answered or valued.

In a course on introductory programming, one lecture introduces the concept of loops. After having explained that all loops need a stop-condition, otherwise they will become infinite loops, one student asked if one also can define a loop inside of another loop. Most students just had heard for the first time about loops, so I decided that this question might be hard to grasp at this moment for most students and to answer it later during the lecture. I valued the question and told this the student ("Good question!"), then wrote down the words "nested loops" on the upper left corner of the whiteboard and told the student that it will be answered later. I then continued with my explanation of loops until we had covered all important parts and did some exercises using REMOTE HAND. I then came back to the question and introduced nested loops and thus answered the student's question.

COLLABORATIVE SUMMARY

You have a list that summarizes (parts of) some elements of the previously covered lecture content or that contains elements the students can think of by themselves based on their prior knowledge or experience.



It keeps the students passive if you just present the list of content covered to the students and run through all the bullets.

Seeing an already complete summarizing list does not require the students to think about the elements of the list, there is *no knowledge reactivation* necessary.

The students also stay *passive* if the list is summarized by the teacher only.

You as teacher might have *no sufficient insight* in what the students remembered and acquired from the previously covered lecture content.



Therefore: Create this list interactively with the help of the students. Write everything down where it is visible to all students. Just after that show the summary you've prepared in advance in order to control is something was missing.

COLLABORATIVE SUMMARY is a good way to improve SUMMARY by actively involving the students in it. This actually works for all kinds of summaries, e.g. lists of some elements they should know about. It also refreshes their old knowledge by making it explicit again. Make sure that all answers are welcome and write also the wrong answers down, but on a separate space (or distribute/delete them later). Discuss with the students why these are wrong answers and make possible connections clear.

CAREFULLY CRAFTED QUESTIONS can support such a COLLABORATIVE SUMMARY, by leading the students the towards the items you want to summarize.

This pattern can also be used for moving towards new content. You need IMAGINATION STIMULATION of the students to think about some new elements they can think of based on their prior knowledge.

It is good to use something for writing down the list which can be changed during the summary, as you might want to also put down unrelated answers and delete them later on after discussing why they don't belong to this list.

An alternative way of creating a COLLABORATIVE SUMMARY is to collaboratively create a diagram that shows the main concepts and the structure of the content of a lesson. These diagrams are known under the name of mind maps or concept maps.

Doing a COLLABORATIVE SUMMARY mostly automatically requires one to CHANGE MEDIA, which might make the learning outcome even better. The only exception is when you're using the same medium during the whole lecture, e.g. only the whiteboard.

A COLLABORATIVE SUMMARY can perfectly combined with ROUND ROBIN so that also the less participative students will be involved in the summary.

Be aware that COLLABORATIVE SUMMARY certainly requires more time than just presenting the list would require. So make sure that you planned enough time for it when doing your LECTURE STRUCTURING.

For a course on software design patterns some previous knowledge is expected from the students. This knowledge includes design principles — like information hiding or low coupling — and also object-oriented concepts — like generalization or interfaces. In the first lecture I prepared this list on a few slides, but didn't show it to the students. Instead I asked the students which design principles they know and why you should apply them, similarly I asked them which object-oriented concepts they know and for what they should be applied. We collaboratively discussed all answers

and I summarized them on a whiteboard. Just after that I used my previously prepared slides to check if we indeed have covered all elements in the list. The slides were then made available to the students for later reference.

2. INVENTORY OF EXISTING PATTERNS

There are quite a few patterns published that directly cover the aspect of introducing activities (in the broadest sense) to lectures. They all can be seen as specialization of ACTIVE STUDENT [Pedagogical Patterns Editorial Board 2012]. Most of them are related to asking questions, whereby two kinds can be distinguished: (1) the teacher is asking questions to trigger students' engagement with the topic and therefore stimulate active participation and (2) the teacher stimulates the students to ask questions, as asking questions is a sign of active participation.

The following list provides an overview of these question-related patterns.

- CAREFULLY CRAFTED QUESTIONS [Larson et al. 2008] - Just asking questions does not automatically help with achieving the desired goals, they better can be carefully crafted upfront in order to engage students more actively and serve the desired goals well.
- SIMPLE ANSWER [Larson et al. 2008] - Asking simple, close ended questions to students engages them in the lesson as they likely know the answers.
- OPEN ENDED QUESTIONS [Larson et al. 2008] - Beside simple questions one can also use questions that require a meaningful answer based on the students' previously acquired knowledge.
- HONOR QUESTIONS [Pedagogical Patterns Editorial Board 2012] - You always should ensure that students are asking questions by honoring them, as question-asking students are actively busy with the lecture content.
- LINE OF REASONING [Larson et al. 2008] - Asking the student also about the thought process her answer is based on stimulates a deeper thinking instead of just giving randomly one possible answer.
- PREGNANT PAUSE [Larson et al. 2008] - If you ask questions then give the students enough time to think about the answer instead of giving the answer immediately yourself.
- UNINTERRUPTED LISTENING [Larson et al. 2008] - Let the students finish their answer, as this will help you with understanding the students' answer in more detail and also shows that you value the whole answer.

The following list summarizes other already described patterns that can be applied to make lectures more interactive:

- DISCUSSIONS WITH PEERS AND STAFF [TELL project output of WP3 2005] - Although on a much higher level, the core of the pattern says that you should consciously include possibilities for discussions among students and teachers.
- MAKE IT THEIR PROBLEM [Schmolitzky 2007] - Introduce a problem which the students collaboratively solve by telling you what to do to solve it. This is closely related to REMOTE HAND.
- MAKE THEM MAKE IT THEIR PROBLEM [Schmolitzky 2007] - Give the students a problem and let them design a small teaching unit for introducing and solving it, which they then have to present to the other students.
- PIECE OF MIND [Larson et al. 2008] - Get (anonymous) feedback from the students by asking them to fill out a card at the end of a lecture with questions they still have, a topic that they haven't understood, or even an "aha" moment they have experienced.
- SIMULATION GAMES [Anthony 1996] - Have some students play a simulation of a complex activity.
- STUDENT DRIVEN LECTURE [Pedagogical Patterns Editorial Board 2012] - Similar to PIECE OF MIND, but here you're collecting the questions at the beginning of the lecture and use them to adjust your lecture so that you can answer or cover the questions appropriately.
- THINK..PAIR..SHARE [Larson et al. 2008] - Ask a CAREFULLY CRAFTED QUESTION, but have the students think about the answer for themselves first and write it down. Then students are pairing and sharing their answers. Finally ask some randomly chosen students to share their partner's answer with the class. This means that everyone is involved, as everyone has to discuss with a partner. They are also encouraged to seriously try to understand their partner's answer, as there is a chance that they have to present this to the group.
- TRY IT YOURSELF [Pedagogical Patterns Editorial Board 2012] - Take a break in your lecture and have the students apply a concept or exercise directly. This way it is easier for them to understand it and easier for you to control if they have understood it.

One important aspect is to include all—or as many as possible—students in the interactions, which is also emphasized by SHOUT IT FROM THE ROOFTOPS. This especially counts for the shy and introvert ones.

There are probably more interactivity patterns, e.g. ones that explore the possibilities of using technology. These still have to be mined.

3. FURTHER REFERENCED PATTERNS

In this section we name all related patterns and their references. For summaries of these patterns we refer to [Köppe 2013b].

- BREAKS [Fricke and Völter 2000]
- BUFFERS [Fricke and Völter 2000]
- BUILD AND MAINTAIN CONFIDENCE [Pedagogical Patterns Editorial Board 2012]
- CHANGE MEDIA [Fricke and Völter 2000]
- CHECK PREREQUISITES [Fricke and Völter 2000]
- DISCOVER YOUR OWN PATTERN [Köppe 2013a]
- EXPAND THE KNOWN WORLD [Pedagogical Patterns Editorial Board 2012]
- EXPOSE THE PROCESS [Pedagogical Patterns Editorial Board 2012]
- IMAGINATION STIMULATION [Köppe and Schalken-Pinkster 2013]
- INVISIBLE TEACHER [Pedagogical Patterns Editorial Board 2012]
- LECTURE STRUCTURING [Köppe and Schalken-Pinkster 2013]
- LET THE PLAN GO [Fricke and Völter 2000]
- LINKING OLD TO NEW [Pedagogical Patterns Editorial Board 2012]
- NAME IS LAST [Fricke and Völter 2000]
- NOBODY IS PERFECT [Pedagogical Patterns Editorial Board 2012]
- OWN WORDS [Pedagogical Patterns Editorial Board 2012]
- PROBLEM ORIENTATION [Fricke and Völter 2000]
- REFLECTION [Pedagogical Patterns Editorial Board 2012]
- ROUND ROBIN [Pedagogical Patterns Editorial Board 2012]
- SHOW IT RUNNING [Schmolitzky 2007]
- SHOW PROGRAMMING [Schmolitzky 2007]
- SHOUT IT FROM THE ROOFTOPS [Pedagogical Patterns Editorial Board 2012]
- SUMMARY [Fricke and Völter 2000]
- TEST TUBE [Pedagogical Patterns Editorial Board 2012]

4. CONCLUSIONS

Even though it is known that lectures have several disadvantages and that alternatives are widely discussed (see e.g. [Grissom 2013]), we still believe that it is possible—up to a certain degree—to include active learning in traditional lectures as well.

In this paper we proposed five educational patterns for lecture design which support the goal of making lectures more interactive and thus improve active learning of the students. All the patterns have been successfully applied several times by the authors and other computer science educators.

In future work we plan to further extend the collection of lecture design patterns. We hereby want to focus on e.g. the technical possibilities of making lectures more interactive, but also on other aspects of good lectures like how to start or finish a lecture in a good way. The final goal is to work towards a pattern language for lecture design.

5. ACKNOWLEDGEMENTS

Big thanks go to our shepherd Eugene Wallingford. His valuable feedback helped a lot to improve this paper. His encouragement showed us that we're on the good path. We also want to thank all students who had to experience these interactivity patterns and provided helpful feedback afterwards. Their reactions also helped to improve the patterns and to evaluate their value from a student's point of view. Finally we are deeply grateful to the participants of the writers' workshop at the PLoP'13 conference for their valuable feedback.

REFERENCES

- ALEXANDER, C., ISHIKAWA, S., AND SILVERSTEIN, M. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press.
- ANTHONY, D. L. G. 1996. Patterns for classroom education. In *Pattern Languages of Program Design 2*, J. Vlissides, J. Coplien, and N. Kerth, Eds. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 391–406.
- FRICKE, A. AND VÖLTER, M. 2000. Seminars - A Pedagogical Pattern Language about teaching seminars effectively. In *Proceedings of the 5th European Conference on Pattern Languages of Programs, EuroPLoP 2000*. Number Dec 12, 2002. Irsee, Germany.
- GAMMA, E., HELM, R., JOHNSON, R., AND VLISSIDES, J. 1994. *Design Patterns: elements of reusable object-oriented software*. Addison-Wesley, Boston, MA.
- GRISSOM, S. 2013. Introduction to special issue on alternatives to lecture in the computer science classroom. *ACM Transactions on Computing Education* 13, 3, 1–5.
- KÖPPE, C. 2013a. A Pattern Language for Teaching Design Patterns. *Transactions on Pattern Languages of Programs* 3, 24–54.
- KÖPPE, C. 2013b. Towards a Pattern Language for Lecture Design: An inventory and categorization of existing lecture-relevant patterns. In *Proceedings of the 18th European Conference on Pattern Languages of Programs, EuroPLoP'13*. Irsee, Germany.
- 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.
- LARSON, K. A., TREES, F. P., AND WEAVER, S. D. 2008. Continuous feedback pedagogical patterns. In *Proceedings of the 15th Conference on Pattern Languages of Programs - PLoP '08*. ACM Press, New York, New York, USA.
- PEDAGOGICAL PATTERNS EDITORIAL BOARD. 2012. *Pedagogical Patterns: Advice for Educators*. Joseph Bergin Software Tools, New York, NY, USA.
- SCHMOLITZKY, A. 2007. Patterns for Teaching Software in Classroom. In *Proceedings of the 12th European Conference on Pattern Languages of Programs (EuroPLoP 2007)*. Irsee, Germany.
- TELL PROJECT OUTPUT OF WP3. 2005. Design patterns for teachers and educational (system) designers. Tech. rep.