CHAPTER 10

Design Patterns for Active Learning

MARY LOU MAHER, NASRIN DEHBOZORGI, MOHSEN DORODCHI, AND STEPHEN MACNEIL

Flipped classrooms, active learning, and peer learning are innovations in education receiving the attention of educational researchers and instructors. Our previous papers describe several strategies for adopting the concept of flipped classrooms in various courses within a computer science education context (Maher, Latulipe, Lipford, & Rorrer, 2015; Latulipe, Long, & Seminario, 2015). As part of our experience with flipped classrooms, we introduced the concept and practice of lightweight teams. The integration of lightweight teams in both introductory computing courses and data structures creates a social learning environment that has led to improvements in academic performance (Latulipe et al., 2015; Latulipe, MacNeil, & Thompson, 2018). In this chapter, we present a more comprehensive view of active learning as pedagogical design patterns, patterns that have emerged from our own practice of active learning.

Active learning has two primary benefits: First, in-class activities create a more engaging learning experience for students, and second, active learning allows for misconceptions to be corrected before assessment (Prince, 2004). Student engagement and collaboration are features of active learning that are often contrasted with a traditional lecture setting where students typically listen to and receive information from the instructor (Prince, 2004). It can be challenging for students to maintain their attention and motivation for the entire lecture period, and many students lose their focus after the halfway point (Köppe & Portier, 2014). Active learning requires students to engage in meaningful learning activities and think about what they are doing (Bonwell & Eison, 1991). These class activities are done either individually or in teams to solve a given problem. This suggests that active learning can be considered as a continuum along which varying amounts of activity can be included throughout a class period.

Although there is some variation in how active learning is defined and discussed, there are some generally accepted definitions that help to distinguish it from nonactive learning (Prince, 2004). Many different types of pedagogy could be classified as active learning, such as team-based learning (TBL) (Smith et al., 2009), cooperative learning (Millis & Cottell, 1997; Feden & Vogel, 2003), collaborative learning, problem-based learning (Prince, 2004), or studio-based learning (Narayanan, Hundhausen, Hendrix, & Crosby, 2012). Although there are instances where students may work on activities alone, active learning usually emphasizes collaboration and learning from peers. Team-based learning (TBL) has the potential to enhance student learning outcomes (LeJeune, 2003) and has been applied across many domains, such as computer science (CS) education (Biggers, Yilmaz, & Sweat, 2009). In many institutions, the classic lecture-style format of teaching is gradually shifting to a practice-based model in which students work in teams while actively and collaboratively developing their understanding of the concepts (Lasserre, 2009). Research in TBL has identified several critical issues related to the successful implementation of teams, including team formation principles, assigning grades to teams, and improving the quality of the experience of working in teams (Michaelsen & Sweet, 2008; Mennecke & Bradley, 1998; Johnson, Johnson, & Smith, 1991). Decisions made by instructors about team formation and grading have an impact on group cohesion and effectiveness. These decisions should not be made solely based on the instructor's intuition but should also consider this research.

Incorporating activities into scheduled class time is a unique opportunity for students to work together under the supervision of an instructor without scheduling conflicts. This type of active learning centers around the social construction of knowledge. Design decisions about team formation, grading, and even the physical structure of the classroom can facilitate or disrupt this social construction of knowledge. In a lecture classroom, desks are often placed in rows to increase seat accessibility and maximize the number of seats that can fit facing forward. In active learning, the physical structure of the classroom can facilitate social interactions, such as placing tables and chairs together so that students can see each other and talk. Given these many design decisions and their effects on learning, best practices and research should be considered by instructors.

Successful implementations of active learning require goal-oriented pedagogical practices based on empirical evidence and research. We present an approach to formalize successful practices in active learning using pedagogical design patterns. Pedagogical design patterns define successful ways to solve recurring problems using a language of problems and solutions, similar to the concept of design patterns in software engineering (Dehbozorgi, 2017). They provide a formalism for capturing emerging successful pedagogical techniques (Dehbozorgi, 2017). Instructors can use pedagogical design patterns as a tool to formulate their teaching practices in a lecture or active learning setting. Many of the existing design patterns in the literature focus on teacher-centered pedagogy and lecturing methods (Dehbozorgi, Maher, MacNeil, & Dorodchi, 2018). Our design patterns serve to fill this gap by focusing on student-centered pedagogy and active learning.

Design patterns help educators share their design ideas in a structured style and also provide a framework for thinking about and comparing design decisions (Preiss, 1999). Design patterns and pattern languages originated in the writings of Alexander et al. on architecture and town planning (Alexander et al., 1977). Alexander et al.'s intention was to democratize architecture and town planning by offering a set of conceptual resources that ordinary people could use in (re)shaping their environment. Their work provides a principled, structured, but flexible resource for vernacular design. In Alexander et al.'s own words, a pattern "describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."

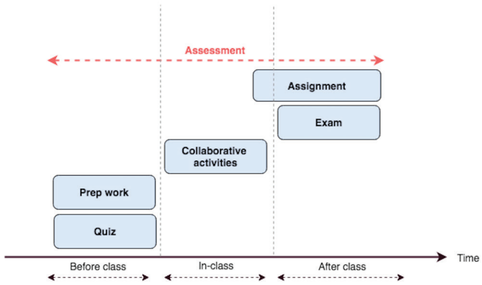
Design patterns in education provide a set of design ideas in a structured format that combines problems with solutions and also offers a rationale that connects research-based evidence with experiential knowledge. Identifying and using design patterns helps instructors encode and apply the knowledge and experience of best practices in education in an iterative and fluid process of designing course materials and activities (Goodyear, 2004).

Identifying a relevant design pattern is the first step in the process of applying that pattern to the practice of teaching. As the number of patterns increases, it becomes harder to find relevant patterns that address a specific problem. In this case, having an object model with multiple attributes may help in indexing the patterns. We have developed an object-based design pattern model that makes explicit the principles of active learning. The core structure of our model is derived from Alexander et al.'s model (Alexander et al., 1977), however, it has been extended to include components and attributes that capture features of active learning and collaboration. The modular structure and defined attributes keep the problem and solution concise. This allows patterns to be easily indexed and allows for the use of concept map representations to show the relationships among patterns. The object-based model representation makes pattern components and their attributes more obvious and cues designers to think about these aspects as they design their course.

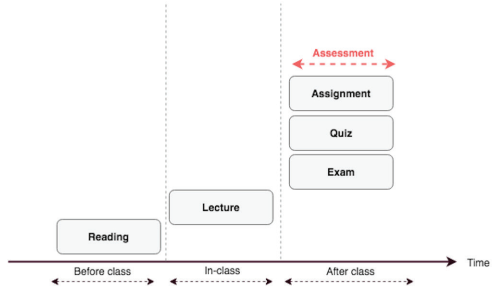
We present our object-based design patterns in active learning that describe problems that occur over and over again. We associate those problems with active learning solutions. Our intent is similar to Alexander et al.'s: the solutions are described as patterns that can be used differently every time to adapt to each classroom's unique context. This provides instructors with the freedom to create their own learning activities and environment. The use of patterns is a way of bridging theory, empirical evidence, experience, and the practical problems of design (Goodyear, 2004). In our case, we are focused on designing course materials and learning activities.

Flipped Classrooms and Active Learning

Flipped classrooms and active learning promote the use of in-class activities for students as an alternative to long lectures and have been successfully implemented in introductory science, technology, engineering, and mathematics (STEM) courses (Hakimzadeh, Adaikkalavan, & Batzinger, 2011; Heines, 2015; Dorodchi et al., 2018). In active learning, the class time shifts from passive learning to active learning. Students are presented with new problems and apply concepts that were introduced prior to coming to class. Class time serves to test student understanding of concepts, address gaps in that understanding, and apply newly learned concepts to increasingly complex problems (Lasserre, 2009). Motivations for creating an active learning classroom are to provide a rich interactive environment, to foster better student engagement, to involve students in collaborative and cooperative problem-solving, and to promote com-



(a) Flow of content delivery and assessment in active learning



(b) Flow of content delivery and assessment in lecture-style

Figure 10.1. Active learning and lecture-based flow of content delivery and assessment.

putational thinking (Hakimzadeh et al., 2011) while socializing and having fun (Dorodchi & Dehbozorgi, 2017). In an active learning classroom, assessment is integrated into all stages of the learning process. This differs from the traditional lecture classroom in which assessment occurs periodically after the lecture and reading, as illustrated in Figure 10.1.

Students typically work in teams in an active learning class to recognize that peer discussion encourages students to come prepared and engages students in explaining while learning. TBL is valuable to student success, even when peers initially did not understand the concepts (Lasserre, 2009). Accountability (positive independence; Johnson et al., 1991) and immediate feedback are two key ingredients of TBL. Accountability can be ensured by evaluating individuals first and teams second. Competition between teams is also used to initialize interest and accountability. Immediate feedback is provided through various means, ranging from traditional materials (cards, scratch tests, color pins, small boards) to technological materials (spreadsheets, in-class scanner, clickers). Students are also encouraged to deepen their understanding by challenging the instructors' questions if they discover errors or ambiguities (Lasserre, 2009).

Teams in Education

We review the literature on teams in education to inform the development of our design patterns for active learning because our approach to active learning has a focus on student engagement and learning from peers. Forming teams that work well together is a hallmark of effective team-based learning. Michaelsen and Sweet's seminal work posits that cultivating cohesion within the team is essential to the success of those teams (Michaelsen & Sweet, 2008). But success in the context of TBL has many definitions. One common way to measure success in teams is to evaluate the quality of the artifact generated by those teams, such as in a capstone course. Examples of these artifacts are group presentations, documentation, and project demonstrations. These artifacts are one important aspect of the TBL experience because in industry these artifacts are highly valued by the company. Teams are formed to maximize the performance of the team in the context of both TBL and professional software engineering. There are a number of guidelines for team formation and composition with this performance goal. Students can be grouped together randomly, by each individual student's preference, or by the instructor when initially forming teams. Randomly formed teams are often preferred because they reduce coalitions (Michaelsen & Sweet, 2008) and homophily (McPherson, Smith-Lovin, & Cook, 2001). A compromise between randomly selected and instructor-selected teams means that teams are chosen algorithmically. For example, CATME attempts to integrate instructor-specified criteria while avoiding scheduling conflicts within teams (Layton, Loughry, Ohland, & Ricco, 2010). The decision about how to form teams is dependent on the purpose of the teams. In active learning, there is a broader set of purposes for teamwork that go beyond the use of teams for project-based learning.

The artifact represents a significant portion of each team member's grade in teams that collaborate on one final artifact. Therefore, teams should be chosen as fairly as possible. This can be challenging because IRATs (individual readiness assessment tests) and other individual performance metrics (such as grade point average [GPA]) only represent one aspect of team performance (Michaelsen & Sweet, 2008). Positive interdependence and individual accountability, which are not accounted for in GPA, are also essential components of collaborative learning (Johnson et al., 1991). Finally, because teams can only deliver one final artifact, they must be able to come to a consensus. For this reason, conflict resolution styles (Forrester & Tashchian, 2013), personality (Peslak, 2006), and leadership styles (Shen, Prior, White, & Karamanoglu, 2007) are sometimes considered when forming software engineering teams.

A broader view of success in the context of TBL might also consider that students construct their professional identities socially within a team. Similarly, students can share metacognitive learning skills, such as techniques for organizing information, test-taking strategies, or problem-solving policies within their teams. Beyond these intended TBL experiences, students who enjoy working together might continue to collaborate after the project or course ends and form informal learning communities.

Different factors should be considered when forming teams to achieve social and collaborative benefits from the team experience. For example, there is a significant negative correlation between teams in which some members have preexisting friendships and performance on a group project (Maldonado, Klemmer, & Pea, 2009). This is one of the reasons that self-selected teams should be avoided; however, these teams can also provide opportunities for students to develop their friendships so that they are more connected to other students in their major (Barker, McDowell, & Kalahar, 2009). Persistence in an academic program has been correlated with a student's sense of social support (DeBerard, Spielmans, & Julka, 2004). So, although some of the factors in forming teams may not lead to high performing teams, they may lead to TBL experiences.

Choosing an appropriate size for teams is task dependent, and recommendations for optimal sizes vary widely in the literature (Adams, 2003). Dyads are a popular choice for CS courses in the form of pair programming. There are two other common recommendations for team size: three to five and five to seven. Three to five is typically recommended for activities that require less structure (Adams, 2003). Larger teams can be considered for more structured interaction. LeJeune recommends five to seven to ensure that the team has enough breadth of skills to complete the task while minimizing social loafing and promoting positive interdependence (LeJeune, 2003). We refer to these two functional sizes as small and medium, respectively. Large can be considered a catchall for other sizes; however, it is generally associated with class-wide activities, such as discussions.

Roles are one way to ensure positive interdependence (Johnson et al., 1991). Assigning roles to students ensures that they work collaboratively and rely on each other. The use of roles has been shown to improve cohesion in programming teams (Mennecke & Bradley, 1998). We identify two types of roles: task-specific and team-specific. Examples of task-specific roles are driver and navigator for pair programming, programmer and documenter in a traditional programming team, and tester. Team-specific roles are designed to keep the team on track. Examples of team-specific roles are timekeeper, encourager, and devil's advocate (Adams, 2003). Roles can be assigned by preference, personality tests, or randomly. Cruz, da Silva, Monteiro, Santos, and Rossilei provide a review of personality tests in software engineering education. The study names Myers–Briggs, Kersey Temperament Sorter, and Neo Five Factor Model as the three most common tests for forming teams (Cruz, da Silva, Monteiro, Santos, & Rossilei, 2011).

It is difficult to assess and grade teams because each individual, and the team as a whole, need to be considered. Grading schemes for assessing individual students often have students share one grade that was assigned to the team, have their individual contributions evaluated, take quizzes to assess individual competencies, or have a cross-validating approach that combines more than one of these schemes (Dehbozorgi, MacNeil, Maher, & Dorodchi, 2018). In Michaelsen and Sweet's original teams, they suggested using individual readiness assurance tests as a way to ensure each student was developing individual competency and cross-validating the individual's contribution to the team (Michaelsen & Sweet, 2008). Peer and self-evaluation are very common methods of assessment because teams may work outside of class time, and the instructor may not be aware of the team dynamics to ensure fair and successful team experiences. Surveys are employed for this kind of assessment, and these surveys can include Likert scales, partner ranking, descriptive word matching, short answers about peers, and journaling about their effort and experiences (Hayes, Lethbridge, & Port 2003). Finally, the weight of the grade is either provided by the instructor in the form of a standard rubric, or the weight of each component is negotiated between the instructor and the students (Michaelsen & Sweet, 2008).

In the next section, we draw on our review of teams in educational settings to describe our model for active learning design patterns that includes TBL.

An Object-Based Model for Pedagogical Design Patterns

Design patterns represent known problems and solutions in a standardized way to enable the sharing of emerging best practices. Design patterns allow designers to research a problem they are currently facing and use practiced solutions rooted in learning theories or empirical rationale. A wide range of pedagogical practices in CS education originates from instructors' expertise. Mapping these pedagogical practices to existing learning theories can be challenging and time consuming for new instructors. Instructors often rely on their intuition or on pedagogies that they have observed in their time as students.

Researchers have adapted different formats for their design patterns. Most pedagogical design patterns in the literature consist of specific parts, such as the "problem" and the "solution" to address that problem. Each set of design patterns has a specific format and language. Our observation of the related research showed that most scholars apply the general structure of patterns proposed by Alexander et al. (1977) regardless of design pattern category. Depending on the context, some researchers used an adapted version of Alexander et al.'s format and added more attributes to the patterns. Most published pedagogical design patterns (Bergin, 2006; Goodyear, 2004; Köppe & Portier, 2013; Köppe & Schalken-Pinkster, 2015; Köppe, 2011) adapt Alexander et al.'s pattern format (Alexander et al., 1977). The beginning section of each pattern is a short summary of the context that explains in what circumstances the solution should be applied, and this is followed by three diamonds. The second part of the pattern includes the problem (in bold) and the forces that shape and refine the problem. These forces that are intended for deeper understanding of the nature of the problem are followed by three diamonds. The next parts of the pattern consist of the solution (in bold), solution details, positive and negative consequences, and a discussion of the possible implementations. Finally, there is the example of pattern implementation that is explained in italics.

Format 1: [Context, <u>Problem</u> {forces}, <u>Solution</u> {solution details}, Positive/negative consequences, Pattern implementation, Examples, Related patterns]

We have developed an object-based design pattern model derived from Alexander et al.'s format (Alexander et al., 1977). This model facilitates selection and adaptation to a new context. Our object-based model emphasizes solutions that include teams to engage students in peer learning. Our model uses dimensions to build on research in TBL (Mennecke & Bradley, 1998; MacNeil, Dorodchi, & Dehbozorgi, 2017; Dehbozorgi, 2017; Dehbozorgi, MacNeil, Maher, & Dorodchi, 2018). Figure 10.2 illustrates this model, its components, attributes, and related values. This model has three main components:

- Pattern name: Describes the general characteristics of the pattern.
- Metadata: Provides high-level information about the pattern. It provides information about the high-level category of the problem this pattern addresses and its goal (e.g., content delivery, assessment or getting students' feedback, individual vs. teamwork, etc.; Smith et al., 2009).
- Pattern core: This component has four main attributes: problem, solution, rationale, and pitfall. The solution includes second-level attributes that are "teamwork" (Smith et

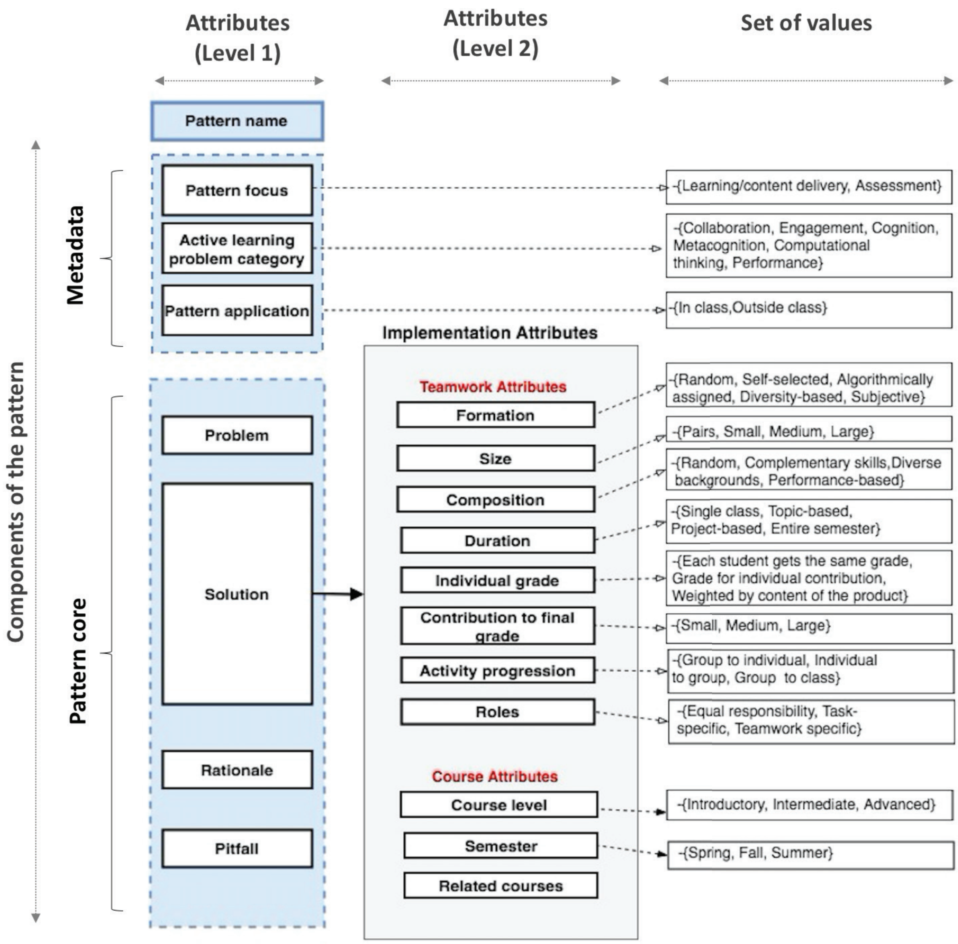


Figure 10.2. Object-based pedagogic design pattern model.

al., 2009) and “course” attributes. The teamwork attributes are: team formation, team size, duration of teamwork, individual grade in teams, teamwork product contribution to final grades, activity progression, and roles in teams. The course attributes provide insights about application of the pattern in a course- or context-specific domain. The course attributes are: course level, semester, and related courses. Depending on how the solution is going to be applied, different values can be assigned to the second level attributes.

Different variations of the teamwork attributes can be practiced in applying the solution. Therefore, several examples of the solution can be provided by setting different values for the teamwork and course attributes. The third attribute of the “pattern core” is the “rationale.” “Rationale” connects research-based evidence with experiential knowledge to justify why the solution is appropriate for the corresponding problem. Design patterns can have unintended

PATTERN NAME: Low-stakes (lightweight) teams in class	
METADATA	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement
Pattern application	In class
PATTERN CORE	
Problem	Students' performance in teams is negatively affected by the importance of the grade.

Solution	Create teams for activities that do not have a significant contribution to final grades and encourage students to learn from each other.
	
	Suggested teamwork attributes (optional*):
	`{formation, size, composition, duration, individual grade, contribution to the final grade, activity progression, roles}`
	Suggested course attributes (optional*):
	`{course level, semester, related CS courses}`

Rationale	Reduces students' stress to perform well to get a good grade and encourages social learning.
Pitfalls	Students may still worry about unequal contribution to teamwork. Students may get discouraged by low grade contribution.

Figure 10.3. Abstract "low-stakes teams in class" pattern class.

or undesirable side effects. This aspect is captured in the fourth attribute as "pitfall" that warns about how the pattern's solution may lead to a different problem that may be addressed by another pattern.

According to the literature, patterns should be "simple and elegant solutions . . . [which] capture solutions that have developed and evolved over time" (Köppe & Schalken-Pinkster, 2013). The intention of the components and attributes in our model is to highlight the pattern details and features. In simpler terms, there is no need for the pattern designer/user to narrate/look for all the details in a verbose pattern description. Instead, this abstract representation is concise and flexible, allowing the practitioners to adopt different variations of attributes while implementing the pattern. Any pattern can have multiple examples of implementation (objects) by setting different combinations of values to teamwork and course attributes. We demonstrate how this model can be used to generate meaningful design patterns for active learning with an example pattern shown in Figure 10.3 and Figure 10.4. The "low-stakes team" pattern has been selected as the abstract representation of the pattern, and two objects are derived from this pattern by setting different values to teamwork attributes. This pattern addresses the problem that students need applied practice with course concepts to go beyond a theoretical understanding they develop during lecture or during prep work. In this example, the only attribute that has the same value in both objects is the "contribution to final grade" that is the basic characteristic of the concept of the low-stakes team.

Lightweight teams Object No. 1:
Teamwork attributes
Formation: `random`
Size: `medium (2-3)`
Composition: `complementary skills`
Duration: `Entire semester`
Individual grade: `grade for individual contribution`
Contribution to final grade: `small`
Activity progression: `group to individual`
Roles: `equal`
Course attributes
Course level: `introductory, intermediate`
Semester: `any`
Related CS courses: `any`

Lightweight teams Object No. 2:
Teamwork attributes
Formation: `self-selected`
Size: `medium (3-4)`
Composition: `diverse background`
Duration: `Entire class`
Individual grade: `weighted by team product`
Contribution to final grade: `small`
Activity progression: `individual to group`
Roles: `task specific`
Course attributes
Course level: `introductory`
Semester: `Fall`
Related CS courses: `programming`

Figure 10.4. Example of two objects derived from the "learning activity in class" abstract pattern.

As shown in Figure 10.3, the concise model clearly addresses students' collaboration and engagement issues. The attributes of "metadata" components provide higher-level information about the pattern. Since this pattern addresses the collaboration issue between students (as read in metadata), the teamwork attributes relate to this solution. By setting different values for teamwork attributes, multiple examples can be generated from a single pattern. Figure 10.4 shows two sample implementations of the "low-stakes teams in class" pattern by assigning values to teamwork attributes in the form of two derived objects from the abstract pattern.

Active Learning Design Patterns

Based on the object model, we have developed 15 patterns focusing on general problems of active learning. These patterns address the problems in four main categories: (a) Prep work patterns, (b) In-class activity patterns, (c) Teamwork patterns, and (d) Reflection and feedback patterns. Below we describe the abstract patterns in each of these four categories using our object-based model.

Prep Work Patterns. The students need to be prepared before coming to active learning classes. The following set of tables shows different patterns and methods of preparation.

In-Class Activity Patterns. Designing meaningful activities in active learning classes can be a challenge for instructors. The following set of patterns give some insights into in-class activities.

Teamwork Patterns. Teamwork is one of the important aspects of active learning environments. Forming teams, balancing time, and assigning grades to students can be a challenge. The following patterns include some guidelines about teamwork in active learning.

Reflection and Feedback Patterns. Frequent feedback and formative assessment is an important aspect of active learning. The feedback can be both about students' experiences during

Table 10.1 Prep Work Pattern: Short Lectures Before Class

Short lectures before class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Engagement/Cognition
Implementation	Outside class
Pattern core	
Problem	Long lectures encourage passive learners, and many students fall asleep in long lectures during class.
Solution	Create short video lectures and make them available online for students to watch before attending a scheduled class activity.
Rationale	Reduce passive learning during class time. Students have more time to ask their questions and get guidance from the instructor during the class.
Pitfalls	Breaking course content into chunks and the process of making a video may be a challenge for the instructor. Students may choose not to watch the video before class. Students may feel that watching videos online and alone is too passive. Watching videos is a form of passive learning that needs a follow-up learning experience.

Table 10.2 Prep Work Pattern: Prep Work Forcing Function

Prep work forcing function	
Metadata	
Pattern focus	Learning/Assessment
Active learning problem category	Cognition
Implementation	Outside class
Pattern core	
Problem	Students might skip doing the prep work before attending the class.
Solution	Do not allow the students to access the in-class material until they have indicated that they have completed the prep work.
Rationale	Students take more responsibility in doing prep work.
Pitfalls	Students might acknowledge that they did the prep material while they have not completely finished the prep work.

Table 10.3 Prep Work Pattern: Short Quiz Before Class

Short quiz before class	
Metadata	
Pattern focus	Learning/Assessment
Active learning problem category	Engagement/Cognition
Implementation	Outside class
Pattern core	
Problem	Students might not pay enough attention to the prep material or skip doing it. Instructors need to know the students' preparedness level before proceeding with the class activities.
Solution	Have students answer a short quiz after completing the prep work.
Rationale	This quiz can act as a forcing function to complete the prep material. The quiz provides an opportunity for students to learn from their mistakes. Based on the quiz grades, instructors can monitor how many have done the prep work and also assess their level of knowledge after finishing the prep work.
Pitfalls	Because of the low grade contribution of prep quizzes, some students might skip doing them. Some students might need additional instruction to learn the content and do well in the quiz even if they have done the prep work. Designing the prequiz with the right challenge level needs to be well thought-out.

Table 10.4 Prep Work Pattern: Collaborative Online Activities Before Class

Collaborative online activities before class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement
Implementation	Outside class
Pattern core	
Problem	Students lack motivation to learn the material and be prepared before the class.
Solution	Design some activities related to the lecture video that students watch before coming to class and have them work collaboratively. Every badge that any individual earns by solving the problems can be rewarded to the whole group.
Rationale	Students are motivated by peer pressure and reward.
Pitfall	Some students might rely on their teammates and not put much effort in solving the problems. Some students may feel that the reward does not have enough direct impact on their grade.

Table 10.5 Prep Work Pattern: Collaborative Online Videos Before Class

Collaborative online videos before class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement
Implementation	Outside class
Pattern core	
Problem	Watching videos alone is passive, and students may get distracted easily.
Solution	Use anchored collaboration techniques to embed forums into video watching sessions. Require that student groups submit a consensus on the most important points of the video lecture before class to get credit for preparation work.
Rationale	Students can interact with their peers and engage more actively as they consume content online.
Pitfall	Determining a student's participation can be a challenge for the instructor.

Table 10.6 In-Class Activity Pattern: Interactive Real-Time Quiz Questions Activity in Class

Interactive real-time quiz questions activity in class	
Metadata	
Pattern focus	Assessment
Active learning problem category	Collaboration/Engagement/Performance
Implementation	In class
Pattern core	
Problem	Students are not always motivated to study preparation materials for the class.
Solution	Develop interactive real-time quizzes that students take during the class. Engage students to answer either individually or with the team. The use of interactive quizzes makes the results visible anonymously to everyone and allows students to see their own and others' mistakes instantly.
Rationale	Engages students in the material with feedback available to them instantly. Helps in learning with low stress. Interactive quizzes are the basis for peer learning while the students are not dependent on their teammates to answer.
Pitfall	Designing quizzes requires time and effort for the instructor. If a student did not do the preparation study, the learning benefit is diminished. Students may not have learned some of the concepts in the preparation study and need more instruction. Most real-time interactive quizzes are multiple-choice questions, and these kinds of questions only address recall, potentially missing application and synthesis learning.

Table 10.7 In-Class Activity Pattern: Applied Learning Activity in Class

Applied learning activity in class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement/Cognition
Implementation	In class
Pattern core	
Problem	Students need to use the concepts from the lecture to learn in more depth and resolve their misunderstandings.
Solution	Expose students to in-class activities performed in small groups that require the knowledge in the preparation work to complete the activity.
Rationale	Students go beyond memorizing generalizations and apply what they are learning. Students figure out if they really understand the material being presented. Students get motivated to do the prep work before coming to class because of the social pressure of working in teams.
Pitfall	Designing class activities and maintaining consistency in the preparation activities with class activities can be a challenge for the instructor. Determining the contribution of the class activities to final grades can also be a challenge for the instructor. Students may not know how to solve problems and will need more time to complete the activity.

Table 10.8 In-Class Activity Pattern: Short Lectures on Demand in Class

Short lectures on demand in class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Cognition
Implementation	In class
Pattern core	
Problem	Students are not able to connect the content of preparation work to a class activity.
Solution	Provide short (5–10 min) lectures during in-class activities that address emerging student misconceptions.
Rationale	Students learn more from minilectures since they are in demand of information and guidance.
Pitfall	Instructors should be careful that the minilectures do not exceed a certain time frame.

Table 10.9 In-Class Activity Pattern: Active Listening Activity in Class

Active listening activity in class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement
Implementation	In class
Pattern core	
Problem	Students need to practice and learn how to listen to other students.
Solution	Ask students in a team to respond to a prompt (Why did you choose this major? How did you answer the quiz question?). Have each student provide a response while the others listen. The listening students are not allowed to interrupt or speak. The student providing the response is given a fixed amount of time to answer. If the student speaking does not need the entire time allocated, then there is silence.
Rationale	Students will learn to listen if they are told not to ask questions or interrupt the student who is speaking.
Pitfall	Identifying the prompt and the amount of time for each student to speak may be difficult.

Table 10.10 Teamwork Pattern: Think-Pair-Share in Class

Think-pair-share in class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement/Cognition
Implementation	In class
Pattern core	
Problem	Group activity can reduce time for individual reflection.
Solution	Structure group activity so that there is time for individual reflection before the group discusses and submits a solution.
Rationale	By providing time for individual reflection and teamwork, different learning styles are accommodated.
Pitfall	Keeping teams on the same schedule is a challenge because students need different amounts of time for reflection.

Table 10.11 Teamwork Pattern: High-Stakes Teams in Class

High-stakes teams in class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement/Cognition
Implementation	In class
Pattern core	
Problem	Students do not demonstrate enough collaborative and social skills to perform well in teams outside the class.
Solution	Assign students to teams during the class and have them work on activities together in senior-level classes.
Rationale	Students learn many concepts from their peers. Class time is more dynamic, and students learn how to work in teams to prepare for being computing professionals.
Pitfall	Students need some time to reflect on concepts individually and not fully rely on teammates to solve problems. Teamwork can impose some grade stress on students (especially high achievers). Fair task distribution in teams and assessing individuals can be a challenge.

Table 10.12 Teamwork Pattern: Low-Stakes (Lightweight) Teams in Class

Low-stakes (lightweight) teams in class	
Metadata	
Pattern focus	Learning/Content delivery
Active learning problem category	Collaboration/Engagement/Cognition
Implementation	In class
Pattern core	
Problem	Students' performance in teams is negatively affected by the importance of the grade.
Solution	Create teams for in-class activities that do not have significant contribution to final grades and encourage students to learn from each other in introductory-level classes.
Rationale	Reduces students' stress to perform well to get a good grade and encourages social learning.
Pitfall	Students may still worry about unequal contribution to teamwork. Students may get discouraged by low grade contribution.

Table 10.13 Teamwork Pattern: Low-Stakes Team Grade Assignment

Low-stakes team grade assignment	
Metadata	
Pattern focus	Assessment
Active learning problem category	Collaboration/Engagement/Performance
Implementation	In class
Pattern core	
Problem	Lightweight teams (Table 10.1) with the grade assigned based on team results disadvantages well-prepared and high-achieving students.
Solution	Assign grade for team activity as the average or the higher of the individual and group grade. This works best with clicker quizzes when you can repoll each question.
Rationale	Encouraging students to come to class prepared.
Pitfall	Low performing students will continue to come unprepared.

Table 10.14 Reflection and Feedback Pattern: Reflection on Teamwork

Reflection on teamwork	
Metadata	
Pattern focus	Assessment
Active learning problem category	Collaboration/Engagement
Implementation	Inside/Outside class
Pattern core	
Problem	Students' reflection on their participation and interaction in teams does not happen unless it is requested.
Solution	Ask students to fill out a short survey (request for reflection) about their teamwork experience after each teamwork activity.
Rationale	Encouraging students to talk about collaborative/cooperative experience encourages learning through self-assessment. This is in contrast to the assessment made by the instructor.
Pitfall	Low-performing students in groups may not provide the necessary details. Students may not appreciate the benefit of reflection and may need to see how it relates to their grade before they take it seriously.

Table 10.15 Reflection and Feedback Pattern: Reflection on Learning

Reflection on learning	
Metadata	
Pattern focus	Assessment
Active learning problem category	Cognition/Metacognition
Implementation	Inside
Pattern core	
Problem	Students do not have many opportunities to reflect on learning in teams, and it does not happen unless it is requested. Instructors are unaware of student challenges in course content; teamwork submission may not reveal each individual's gap in understanding the class material.
Solution	Have students fill out a two-question survey before leaving the class: "What did you learn in this class?" and "What was the most challenging concept for you in this class?"
Rationale	Develops metacognitive skills because it encourages students to think about their learning. Improves learning through self-assessment in contrast to the assessment made by the instructor. The student responses to these two questions can also help the instructor understand what the students found challenging. Reflection is super short so students are more likely to do it.
Pitfall	Asking students directly about their learning may not always reveal valuable information. Some students may not answer the forms thoughtfully.

the active learning class and working with teams or about their learning process (metacognition) and concepts they learned or did not learn in class. Depending on the course level, the required assessment can be different. For example, in capstone courses, teamwork has a higher contribution to students' final grades than in introductory courses. The following patterns are examples of reflection and assessment practice in active learning.

Relating Patterns Using Concept Maps (Pattern Language)

Evaluating individual design patterns can be challenging because each pattern addresses an instructional problem encountered in the classroom. However, concept mapping is a novel approach when considering how patterns interact and may be used to locate potential pitfalls that could occur while following instructional designs. According to Alexander et al. (1977), combining patterns provides a more valuable outcome to overall instruction technique; it further develops pattern language, a method of describing instructional design practices. In education, developing good instructional pattern language is crucial to addressing various student, instructor, and classroom needs. Existing patterns are largely based on Alexander et al.'s approach: design space encoded in narrative. In our study, we utilize and build on similar concept mapping to visualize relationships between object-based pedagogical structures.

In Figure 10.5, we show an object-based representation of the relationships between several patterns. Each pitfall leads to an existing or corrective pattern; the corrective pattern then addresses that pitfall as a problem. For example, a short lecture before class contains two pitfalls: students may not have learned some concepts during preparation for class, and students may not have watched the lectures or other information given prior to class. These two pitfalls then lead to possible corrective patterns: in-class lectures on demand and in-class, interactive, real-time quiz activities. However, these corrective patterns also contain their own associated pitfalls.

In our concept map, we address pitfalls associated with both collaborative and noncollaborative learning. Note that Figure 10.5 only simplifies a larger body of instructional patterns while providing an overview of how multiple patterns may or may not fit together. These overviews can lead to a more holistic understanding of the decisions made when designing classroom experiences. Likewise, links between patterns are not prescribed or absolute; they serve as suggested pathways through the design space.

In object-based patterns, all relationships are directional and are described in pitfall sections; this supports the idea that potential complications are important components of instructional design patterns. Conversely, in narrative formats, relationships between patterns are defined by upper case KEY_WORDS that are integrated into the narration; the reader is then left to identify relationships between patterns by reading the pattern narrative.

In the Student **Minors** pattern (Köppe & Schalken-Pinkster, 2013), a similar narrative format, relationships between patterns are largely implied. The names of related patterns within the structure are not always descriptive, making the relational interpretation even more challenging. In our study, we attempt to identify the types of relationships mentioned in the solution of the Student Minors pattern.

According to our research, there are four types of patterns in the Student **Minors** model: originating patterns, similar patterns, course-specific patterns, and related patterns. Throughout the model, we found that relationship diversity tends to make pattern language less consistent and interpretable in this model.

In the object-based model, we introduce the idea of specific attributes, which eliminates the needs for similar patterns or course-specific patterns in the pattern language. We include solutions to various classroom problems by developing examples of abstract patterns, which have different values assigned to the pattern core and avoid redundancy. In the resulting object-based pattern language, we have a hierarchy of problem–solution pairs, including those associated with active and collaborative learning, that generate pitfalls directed to other patterns as possible solutions.

Active learning and collaboration are often coupled with flipped classrooms. Flipped class-

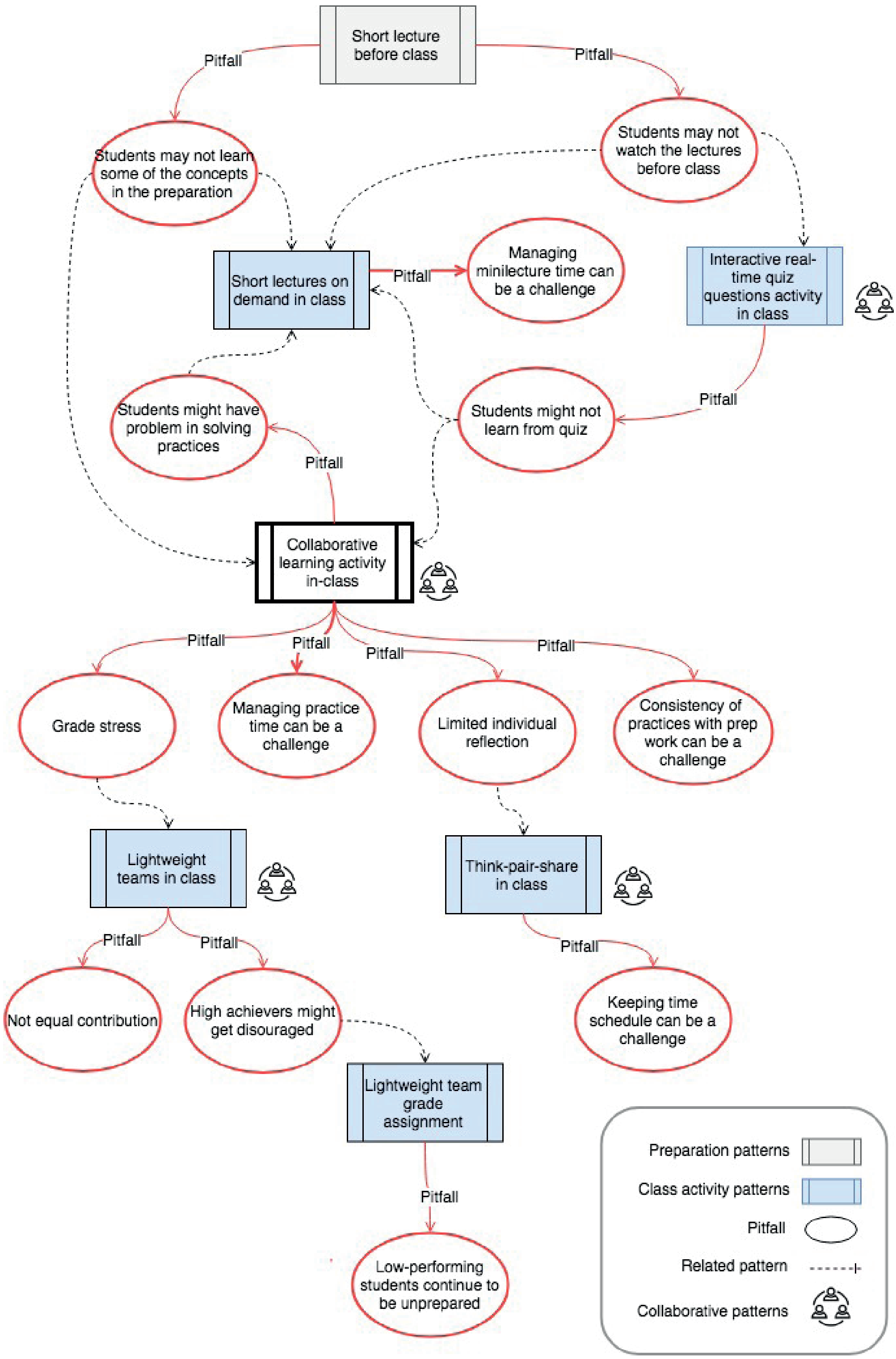


Figure 10.5. An object-based representation of the relationships between several patterns.

rooms provide students with an opportunity to become familiar with the materials at home, get practice with it in class, and then extend their understanding through after-class assignments. This complexity means that implementing one single pattern would not likely be sufficient to create a successful collaborative learning environment. Instead, multiple patterns should be combined to consider the many problems faced by instructors and to account for potential challenges that are an unavoidable part of any pedagogical technique. This aspect highlights the importance of a usable and comprehensive pattern language that can be applied by various designers.

Our object-based pattern language with concept map representation describes how a sequence of patterns can be combined and applied together. Narrative lecture-based patterns, however, suggest several patterns are related, but they can be verbose and difficult to interpret. As the number of patterns grows, the narrative format can become overwhelming. Ultimately, an object-based approach makes it easier to find the pattern that you need and apply it with confidence, knowing that pitfalls can be mapped to other patterns.

Evaluation of Design Patterns

Our design patterns were evaluated in two ways: (a) by measuring the frequency that patterns were applied, and (b) by evaluating how the application of these designs impact CS faculty. To collect the data necessary to evaluate these two aspects, we held a series of active learning design pattern workshops.

In May 2016, we held a summer institute for active learning at our college with 15 participating faculty. The purpose of this institute was to share the design patterns that we were developing. During the institute, we presented several of our emerging design patterns and our object-based model for structuring active learning design patterns. The following semester, those faculty members adopted the patterns in their classes.

This workshop provided us with feedback about what was happening in the classroom and helped us adapt our patterns to the many problems and solutions that were emerging in our instructors' classrooms. To get a better sense for the existing active learning pedagogies that are being practiced at our college, we conducted a second summer institute in May 2017; 19 faculty members attended. We asked instructors to self-report which active learning design patterns they used in their classes during the previous academic year.

The "Learning Activity in Class" and the "Lightweight Teams in Class" patterns were the two most commonly used patterns. The "Teamwork in Class" pattern was not developed at the time of this summer institute. This explains why data were not provided for this pattern application at that time. The list of patterns with resulting application frequency data is presented in Figure 10.6.

To evaluate how the application of these patterns impacted the teaching practices of faculty, we distributed a survey to the participants. We also conducted a focus group discussion in May 2018. The research question that we wanted to answer for this evaluation was "How does the application of design patterns impact the practice of active learning?" To answer this research

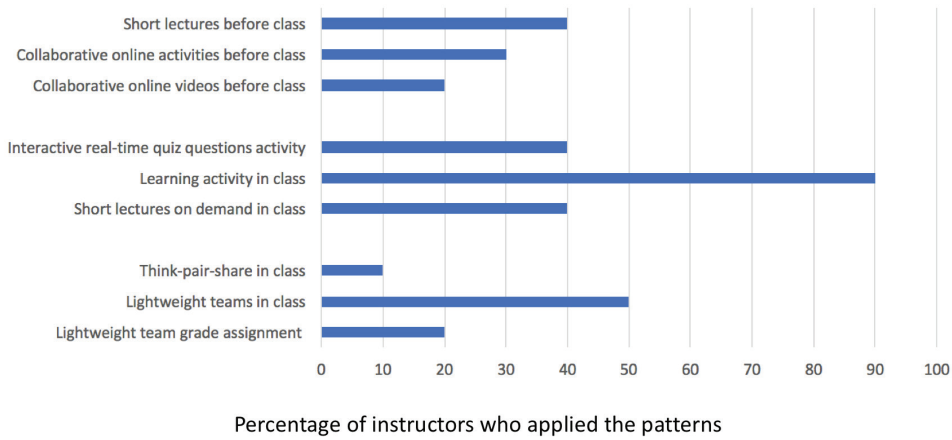


Figure 10.6. Application frequency of active learning design patterns during 2016.

question, we reached out to faculty in our college. Our college has approximately 30 CS faculty members that engage in active learning. We introduced our design patterns and the concept maps to these faculty members though digital handbooks. We shared updates to these patterns continually from 2016 to 2018. In this study, 21 of the 30 faculty members responded to a survey. The survey includes two sections, consisting of four five-point Likert scale responses and two open-ended questions. The four response statements probed the faculty members' experiences when they applied the design patterns in their classes. We asked about their agreement with the following statements:

- I found the use of design patterns helpful in my active learning classes.
- Applying design patterns has improved my active learning teaching experience (i.e., for providing material, managing class time, etc.).
- These design patterns have helped me develop a more structured approach to active learning (i.e., raising awareness of team formation, problems and solutions, pitfalls, etc.).
- One or more design patterns has provided new insights and ideas for my teaching.

The following open-ended questions were developed to capture faculty interest in applying or not applying our research patterns in the future. We also encouraged the faculty to provide examples of how design patterns improved active learning practices in their classrooms.

- Question 5: Which pattern(s) are you planning to use during fall 2018? (mark all that apply).
- Question 6: Provide some examples on how design patterns can improve active learning (AL) practices.

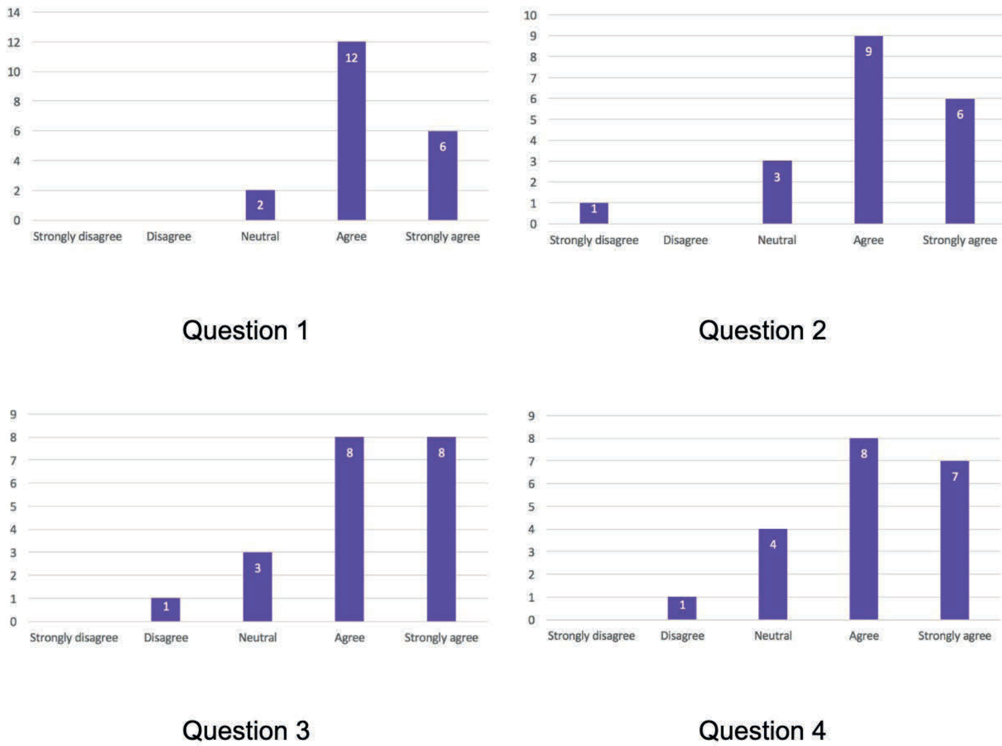


Figure 10.7. Distribution of answers about the impact of design pattern on active learning practices.

Faculty responded to the four statements using a five-point Likert scale (*Strongly disagree* = 1, *Strongly agree* = 5). The distribution of answers is presented in Figure 10.7. These results show that most (76%) of the faculty members who responded to the survey support the benefits of design patterns in their practices of active learning (Figure 10.8).

Figure 10.8 presents the result of Question 5, which shows that the top two patterns selected by faculty were "Learning Activity in Class" (90%) and "Teamwork in Class" (90%). The result is somewhat consistent with pattern application frequency analysis that we performed one year before in 2017. In the 2017 summer institute, the teamwork pattern was not developed; the only pattern that denoted teamwork in class was "Lightweight Teams." This pattern was also one of the top two most frequently used patterns that year. The second one was "Learning Activity in Class," which again ranked highest in the 2018 study. These popular patterns are related to incorporating opportunities for collaboration into the classroom.

We conducted a thematic analysis of the answers to Question 6. Figure 10.9 shows the identified themes and patterns in Question 6 answers. Four of the seven themes related to providing more structure for classroom activities and for novice instructors. Having design patterns can help new instructors get up and running with the current best practices in our college. This can save them time, but it also prevents them from reverting to their familiar and comfortable

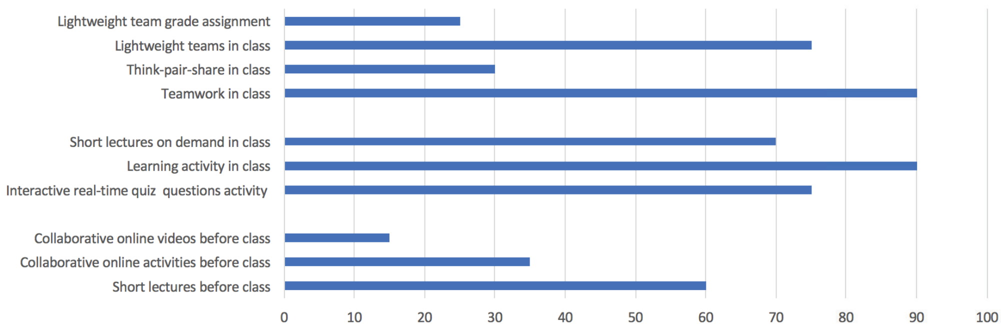


Figure 10.8. Distribution of answers about application of the patterns.

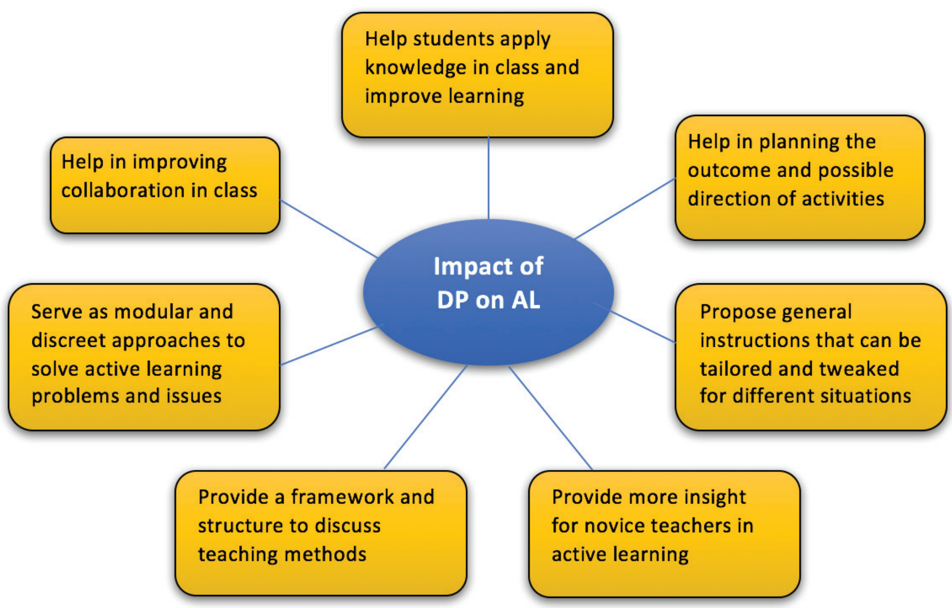


Figure 10.9. The result of thematic analysis on "Impact of Design Pattern (DP) on Active Learning (AL)" dataset.

lecture-based teaching practices. Helping to improve collaboration was also a theme here that reiterates the responses received for our 2018 rankings of which design patterns were most adopted, with the top two being related to collaboration.

We also conducted a focus group with eight CS faculty members who applied design patterns in their active learning practices. The goal of the focus group was to discuss their concerns, satisfaction, practices, and understanding of active learning and the role that design patterns play in their practice of active learning. One part of this focus group was dedicated to

Figure 10.10. Focus group active learning and design pattern word cloud.

an activity in which we asked participants to write their perceptions about active learning. We also asked them to explain design patterns in single words separately on sticky notes and stick them on the wall. After all of the participants had done so, we asked the participants to talk about what they wrote and share their insights with others.

Keywords collected from the participants are illustrated in the form of a word cloud in Figure 10.10. In this presentation, the most frequently used keywords are larger in size. The result shows that the faculty members associated active learning keywords such as *Engagement*, *Collaborative*, *Effective*, and *Learning* more frequently; they related the concept of design patterns with keywords such as *Framework* and *Structure*. Our analysis of this activity shows that the design patterns serve the purpose of providing a framework and a structure for their active learning practice. Results from the focus group indicate that design patterns are achieving their goal of communicating the successful practice of active learning in a structured way that can be applied by faculty.

Conclusion

In this chapter, we present active learning design patterns that have emerged from our practice of active learning in CS education. We developed an object-based design pattern model that captures the attributes of active learning combined with team-based learning. As part of our research, we applied our object-based design patterns in various classrooms with diverse numbers of students and various course topics. Understanding that design patterns can facilitate developing an active learning practice, it was important for us to provide a holistic solution to classroom pedagogy, one that considers inherent limitations and potential complications that any one pedagogical technique may face.

Throughout our study, we were able to identify design patterns for active learning techniques that focus on peer learning. We developed an object-based model in contrast to the narrative format to avoid the difficulty in identifying the links between problems, solutions, and pitfalls in related patterns. We claim that explicit dimensions that capture pitfalls can assist in avoiding potential complications.

Our object-based design patterns simplify problems and solutions by making patterns more readable. They highlight dimensions that are important for team-based active learning, such as team formation, size, roles, and grade weight. They also highlight the importance of readability, flexibility, and functionality in diverse classroom environments. We are confident that instructors can more easily apply pedagogical patterns for active learning with this model leading to more successful student learning.

Acknowledgments

This work is supported by the National Science Foundation Award 1519160: IUSE/PFE: RED: The Connected Learner: Design Patterns for Transforming Computing and Informatics Education.

References

Adams, S. G. (2003). Building successful student teams in the engineering classroom. *Journal of STEM Education Innovations and Research*, *4*, 1–6.

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A pattern language: Towns, buildings, construction*. Oxford, England: Oxford University Press.

Barker, L. J., McDowell, C., & Kalahar, K. (2009). Exploring factors that influence computer science introductory course students to persist in the major. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)* (pp. 153–157). New York, NY: Association for Computing Machinery (ACM).

Bergin, J. (2006). Active learning and feedback patterns: Version 4. *Proceedings of the 2006 Conference on Pattern Languages of Programs (PLoP '06)* (article 6). New York, NY: Association for Computing Machinery (ACM).

Biggers, M., Yilmaz, T., & Sweat, M. (2009, March). Using collaborative, modified peer-led team learning to improve student success and retention in intro CS. *ACM SIGCSE Bulletin*, *41*(1), 9–13.

Bonwell, C. C., & Eison, J. A. (1991). *Active learning: Creating excitement in the classroom* (ASHE-ERIC Higher Education Report No. 1). Washington, DC: The George Washington University, School of Education and Human Development.

Cruz, S. S., da Silva, F. Q., Monteiro, C. V., Santos, P., & Rossilei, I. (2011). Personality in software engineering: Preliminary findings from a systematic literature review. *Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference* (pp. 1–10). IET.

DeBerard, M. S., Spielmans, G. I., & Julka, D. L. (2004). Predictors of academic achievement and retention among college freshmen: A longitudinal study. *College Student Journal*, *38*(1), 66.

Dehbozorgi, N. (2017). Active learning design patterns for CS education. *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 291–292). New York, NY: Association for Computing Machinery (ACM).

Dehbozorgi, N., Maher M. L., MacNeil, S., & Dorodchi, M. (2020). An object-based pedagogical design pattern model for collaborative active learning. (manuscript to be submitted).

Dehbozorgi, N., MacNeil, S., Maher, M. L., & Dorodchi, M. (2018, October). A comparison of lecture-based and active learning design patterns in CS education. *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). IEEE. doi:10.1109/FIE.2018.8659339

Dorodchi, M., & Dehbozorgi, N. (2017, June). Addressing the Paradox of Fun and Rigor in Learning Programming. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (pp. 370-370). doi:10.1145/3059009.3073004

Dorodchi, M., Benedict, A., Desai, D., Mahzoon, M. J., MacNeil, S., & Dehbozorgi, N. (2018, October). Design and Implementation of an Activity-Based Introductory Computer Science Course (CS1) with Periodic Reflections Validated by Learning Analytics. In 2018 IEEE Frontiers in Education Conference (FIE) (pp. 1-8). IEEE.

doi:0.1109/FIE.2018.8659196

Feden, P. D., & Vogel, R. M. (2003). Methods of teaching: Applying cognitive science to promote student learning. New York, NY: McGraw-Hill Humanities, Social Sciences & World Languages.

Forrester, W. R., & Tashchian, A. (2013). Effects of personality on conflict resolution in student teams: A structural equation modeling approach. *Journal of College Teaching & Learning* (Online), *10*(1), 39.

Goodyear, P. (2004). Patterns, pattern languages and educational design. In R. Atkinson, C. McBeath, D. Jonas-Dwyer, & R. Phillips (Eds), *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference* (pp. 339–347). Retrieved from http://www.ascilite.org.au/conferences/perth04/procs/goodyear.html

Hakimzadeh, H., Adaikkalavan, R., & Batzinger, R. (2011, November). Successful implementation of an active learning laboratory in computer science. In Proceedings of the 39th annual ACM SIGUCCS conference on User services (pp. 83-86). doi:10.1145/2070364.2070386

Hayes, J. H., Lethbridge, T. C., & Port, D. (2003, May). Evaluating individual contribution toward group software engineering projects. In 25th International Conference on Software Engineering, 2003. Proceedings. (pp. 622-627). IEEE. doi:10.1109/ICSE.2003.1201246

Heines, J. M., Popyack, J. L., Morrison, B., Lockwood, K., & Baldwin, D. (2015, February). Panel on flipped classrooms. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 174-175). doi:10.1145/2676723.2677328

Johnson, D. W., Johnson, R. T., & Smith, K. A. (1991). *Active learning: Cooperation in the college classroom*. Edina, MN: Interaction Book Company.

Köppe, C. (2011). Continuous activity: A pedagogical pattern for active learning. *Proceedings of the 16th European Conference on Pattern Languages of Programs (EuroPLoP '11)* (article 3). New York, NY: Association for Computing Machinery (ACM).

Köppe, C., & Portier, M. (2014). Lecture design patterns: Improving the beginning of a lecture. *Proceedings of the 19th European Conference on Pattern Languages of Programs* (p. 16). New York, NY: Association for Computing Machinery (ACM).

Köppe, C., & Schalken-Pinkster, J. (2013, October). Lecture design patterns: improving interactivity. In *Proceedings of the 20th Conference on Pattern Languages of Programs* (pp. 1-15). doi:10.5555/2725669.2725697

Köppe, C., & Schalken-Pinkster, J. (2015). Lecture design patterns: Laying the foundation. *Proceedings of the 18th European Conference on Pattern Languages of Program (EuroPLoP '13)* (article 4). New York, NY: Association for Computing Machinery (ACM).

Lasserre, P. (2009, May). Introduction to team-based learning. In *Proceedings of the 14th Western Canadian Conference on Computing Education* (pp. 77-78). doi:10.1145/1536274.1536296

Latulipe, C., Long, N. B., & Seminario, C. E. (2015, February). Structuring flipped classes with lightweight teams and gamification. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 392-397). doi:10.1145/2676723.2677240

Latulipe, C., MacNeil, S., & Thompson, B. (2018, October). Evolving a Data Structures Class Toward Inclusive Success. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE. doi:10.1109/FIE.2018.8659334

Layton, R. A., Loughry, M. L., Ohland, M. W., & Ricco, G. D. (2010). Design and validation of a web-based system for assigning members to teams using instructor-specified criteria. *Advances in Engineering Education*, *2*(1), 1–28.

LeJeune, N. (2003). Critical components for successful collaborative learning in CS1. *Journal of Computing Sciences in Colleges*, *19*, 275–285.

MacNeil, S., Dorodchi, M., & Dehbozorgi, N. (2017, October). Using spectrums and dependency graphs to model progressions from introductory to capstone courses. In 2017 IEEE Frontiers in Education Conference (FIE) (pp. 1-5). IEEE. doi:10.1109/FIE.2017.8190599

Maher, M. L., Latulipe, C., Lipford, H., & Rorrer, A. (2015). Flipped classroom strategies for CS education. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 218–223.

Maldonado, H., Klemmer, S., & Pea, R. D. (2009). When is collaborating with friends a good idea? Insights from design education. *Proceedings of the 9th International Conference on Computer Supported Collaborative Learning (CSCL '09), International Society of the Learning Sciences*, *1*, 227–231.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 415–444.

Mennecke, B., & Bradley, J. (1998). Making project groups work: The impact of structuring group roles on the performance and perception of information systems project teams. *Journal of Computer Information Systems*, *39*(1), 30–36.

Michaelsen, L. K., & Sweet, M. (2008). The essential elements of team-based learning. *New Directions for Teaching and Learning*, *116*, 7–27.

Millis, B. J., & Cottell Jr., P. G. (1997). *Cooperative learning for higher education faculty* (Series on Higher Education). Phoenix, AZ: Oryx Press.

Narayanan, N. H., Hundhausen, C., Hendrix, D., & Crosby, M. (2012). Transforming the CS classroom with studio-based learning. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 165–166). New York, NY: Association for Computing Machinery (ACM).

Peslak, A. R. (2006). The impact of personality on information technology team projects. In *Proceedings of the 2006 ACM SIGMIS CPR Conference on Computer Personnel Research: Forty-Four Years of Computer Personnel Research: Achievements, Challenges and the Future* (pp. 273–279). New York, NY: Association for Computing Machinery (ACM).

Preiss, B. R. (1999). Design patterns for the data structures and algorithms course. *SIGCSE '99: The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science*, 95–99.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, *93*(3), 223–231.

Shen, S. T., Prior, S. D., White, A. S., & Karamanoglu, M. (2007). Using personality type differences to form engineering design teams. *Engineering Education*, *2*(2), 54–66.

Smith, M. K., Wood, W. B., Adams, W. K., Wieman, C., Knight, J. K., Guild, N., & Su, T. T. (2009). Why peer discussion improves student performance on in-class concept questions. *Science*, *323*(5910), 122–124.

Vlissides, J., Helm, R., Johnson, R., & Gamma, E. (1995). Design patterns: Elements of reusable object-oriented software. *Reading: Addison-Wesley*, *49*(120), 11.

Wilkins, D. E., & Lawhead, P. B. (2000). Evaluating individuals in team projects. In S. Haller (Ed.), *Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education (SIGCSE '00)* (pp. 172–175). New York, NY: Association for Computing Machinery (ACM).