A Comparison of Lecture-based and Active Learning Design Patterns in CS Education

Nasrin Dehbozorgi
Department of Computer Science
UNC Charlotte
Charlotte, USA
Ndehbozo@uncc.edu

Stephen MacNeil
Department. of Computer Science
UNC Charlotte
Charlotte, USA
Smacnei2@uncc.edu

Mary Lou Maher
Department. of Software & Information Systems
UNC Charlotte
Charlotte, USA
M.maher@uncc.edu

Mohsen Dorodchi
Department. of Computer Science
UNC Charlotte
Charlotte, USA
Mohsen.dorodchi@uncc.e

***Abstract*— This paper describes and compares two categories of pedagogical design patterns that have emerged from CS education practice: lecture-based design patterns and active learning design patterns. Pedagogical design patterns provide faculty with combinations of generalized descriptions of problems and solutions that occur in teaching and learning. The benefit of forming design patterns is the codification of successful practice that can be reused in multiple scenarios and draw on the creativity of the instructor for defining the details relevant to the course and the students. Design patterns have been represented in many formats since Alexander's initial design pattern model highlighting different aspects of what is important in each domain in which the patterns are created and used. This paper analyzes design patterns emerging from recent developments in lecture-based pedagogy and active learning in CS education. Traditional lectures in computer science, engineering, and other STEM disciplines are being reconsidered due to research that shows that students are less likely to learn while listening and more likely to learn while actively engaged. Design patterns that address problems and provide potential solutions to traditional lectures in computer science education have been published that provide solutions to engage students during the lecture. The pedagogy of flipped classrooms and active learning have recently been adopted by many faculty in Computer Science leading to emerging design patterns for active learning. We compare how previously published lecture-based patterns and our active learning patterns address similar problems with different solutions to engaging students. We show how an object-based structure for pedagogical design patterns can provide additional information about the problems and the solutions addressed by the patterns that are more easily indexed and combined.**

Keywords—*pedagogical design patterns, object-based pattern model, pattern language, concept map, CS education, social construction of knowledge*

I. Introduction

Computer science students are expected to graduate with knowledge of computer science topics and the skills to apply this knowledge to pedagogical and real world problems. Active learning affords students the opportunity to understand and apply course topics during class time in the presence of the instructor and teaching assistants.

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.

Active learning has two primary benefits: misconceptions can be corrected before assessment and in-class activities create a more engaging learning experience for students [2]. Student engagement and collaboration are features of active learning that are often

contrasted by traditional lecture setting where students passively receive information [2]. Active learning is often used because it requires students to engage in meaningful learning activities and think about what they are doing [3].

It can be challenges for students to maintain their attention and motivation for the entire class period and many students may start to lose their focus after the halfway point of a long lecture [1]. This has been a motivating factor for integrating more activities into lectures so that students remain engaged throughout the lecture. These class activities are done either individually or in teams to solve a given problem. This indicates that active learning can be considered as a continuum along which varying amounts of activity can be included in a class period.

Although there is some variation in terms of how active learning is defined and discussed, there are some generally accepted definitions which helps to distinguish it from non-active learning [2]. There are many different types of pedagogy which could be classified as active learning, such as team-based learning (TBL) [5], cooperative learning [6,7], collaborative learning [2], problem based learning [2] or studio-based learning [8]. Although there are instances where students may work on activities alone, many cases of active learning have an emphasis on collaboration and learning from peers. Incorporating activities during scheduled class time is a unique opportunity where students can work together without schedule conflicts under the supervision of an instructor. This type of active learning centers around the social construction of knowledge. The physical structure of the classroom can facilitate this social aspect, such as by placing chairs and tables in such a way that they are conducive to collaboration.

Successful implementations of active learning requires well-studied, goal-oriented pedagogical practices that are based on empirical evidence and research. We present an approach to formalize successful pratice 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 [18]. They provide a

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

formalism for capturing emerging successful pedagogical techniques [17]. Instructors can use pedagogical design patterns as a tool to formulate their teaching practices either in a lecture or active learning setting. There are many design patterns in the literature that focus on different aspects of pedagogy, most of which are teacher-centered pedagogy and lecturing methods [4].

Design patterns are first and foremost a way for designers to implement solutions to known problems. 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 mutiple attributes may help in indexing the patterns. In this paper, 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's model [10], however, it has been extended to include components and attributes that capture features of active learning and collaboration. The modular structure of the model and defined attributes keep the problem and solution concise, allowing patterns to be easily indexed, and allow 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 cue designers to think about these aspects as they design their course.

In this paper, we present our object-based design pattern model and compare it to existing narrative design pattern models. To highlight some of the differences between active learning and lecture-based patterns we compare this design pattern representation to the narrative form used in previously published lecture-based patterns. Finally, we discuss what these models say about the differences between active learning and lecture-based pedagogy.

II. Pedagogical Design Patterns in CS Education

Design patterns represent known problems and solutions in a standardized way to enable sharing emerging best practices. Design patterns allow designers to look up a problem that they are currently facing and use practiced solutions which are often rooted in learning theories or empirical rationale. There is a wide range of pedagogical practices in CS education that originate from instructors' expertise. Mapping the experience and practice to the theories of learning and motivation is not easy, especially for new instructors. Design patterns provide a framework to formalize this connection between problems and existing solutions based on theories or experience.

We have conducted a review of 235 existing pedagogical patterns in published papers in 18 different venues [4]. We grouped these patterns into 6 main categories by looking for emergent themes in the problems that the patterns addressed. First we identified themes based on the similarities found in the problems identified in the patterns, then we associated each pattern with one or more themes to form groups of patterns. After identifying a group for each pattern, we reviewed the patterns in each group to confirm that the pattern is in the most relevant category. In cases where a pattern falls into two categories, it was placed under the category that described the core of the problem that it addressed. The six themes that we identified based on these existing patterns are: lecture design, feedback and assessment, course design, diversity imbalance, teamwork and collaboration, assignment and class activities. The number of patterns in each category were counted and percentage was calculated. As shown in Fig. 1, the majority of published patterns address common problems related to improving the value of lectures (74%) and the least number of patterns address problems related to assignment and class activities (2.1%) or students' collaboration and teamwork problems (3%).

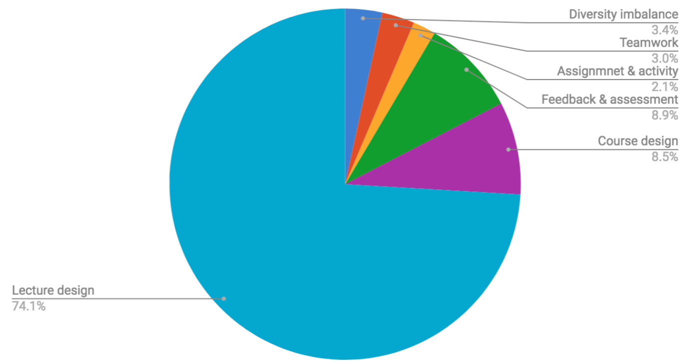

Fig. 1. The distribution of pedagogical patterns

The pedagogical design patterns that we have reviewed previously are presented in different formats but each are rooted in the format presented by Alexander [10]. All patterns include a 'problem' and 'solution'. However, depending on the context some patterns included additional attributes. Below we show five variations of the Alexander's format applied in the patterns we reviewed [4].

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

Format 2: *[Summary, Context, <u>Problem</u> {forces}, <u>Solution</u> {solution details}, Positive/negative consequences, Pattern implementation, Examples]*

Format 3: *[<u>Problem</u> {context, force}, <u>Solution</u>, Rationale, Examples, Consequences]*

Format 4: *[Context, <u>Problem</u> {forces}, <u>Solution</u>, Implementation, Consequences, Examples]*

Format 5:*[Context, <u>Problem</u> {forces}, <u>Solution</u>, Rationale, Examples, Related patterns]*

One of the commonalities that we observed across all of the CS pedagogical design patterns that we reviewed is that they present the problems and solutions in a narrative form, similar to Alexander's design patterns. In this work, we have developed an object-based pattern model which has Alexander's format at its core, but have extended it to include components and attributes which capture features that distinguish the patterns based on their emphasis on active learning and team-based learning.

III. OBJECT-BASED DESIGN PATTERN MODEL

We have created an object-based design pattern model is derived from Alexander's format [10], our practices of active learning over 4 years, and a review of research on team based learning [15,16, 4]. Fig. 2 illustrates this model, its components, attributes and related values.

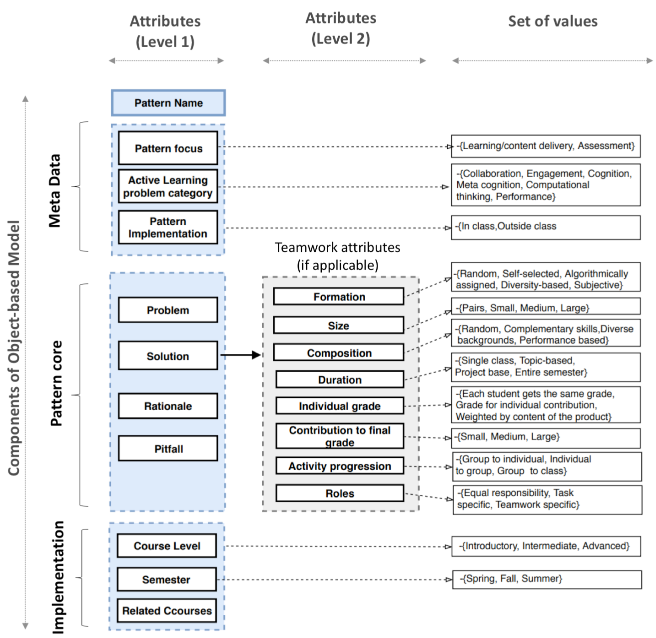

Fig. 2. Developed object-based pedagogic design pattern model

This model has four main components: 'pattern name', 'meta-data', 'pattern core', and 'implementation'. The 'pattern name' describes the general characteristics of the pattern whereas the 'meta-data', 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.) [4]. The 'pattern core' component, has four main attributes: problem, solution, rationale, and pitfall. Because the problem and solution are paired to describe the general issue and how it is going to be addressed by the pattern, the solution also includes second level attributes which capture the collaborative aspect of the solution (when applicable). These second-level 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 [4]. 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 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 or undesireable side-effects. This aspect is captured in the 'pitfall' attribute which warns about how the pattern's solution may lead to a different problem which may be addressed by another pattern. Finally, the 'implementation' component of the model provides insights about application of the pattern in a course or context specific domain. This part includes three attributes of: course level, semester and related courses.

According to literature, patterns should be "simple and elegant solutions . . . [which] capture solutions that have developed and evolved over time" [13]. The intention of the developed components and attributes in our model is to highlight the pattern details and features. In other words, there is no need for the pattern designer/user to narrate/look for all the details in a very verbose pattern description. Instead, this abstract representation is concise and flexible allowing the practitioners to adopt different variations of attributes in implementing the pattern.

Based on the object model, we have developed 10 patterns mainly focusing on general problems of active learning such as students' preparation before class and collaborative in-class activities. Since the contributions of this work is mainly about the developed object-based model and how it captures the active learning features, we are presenting only one of the patterns as an example in this work.

In the process of developing this model and the generated patterns, we held three workshops with faculty to collect their insights and identify the emerging design patterns based on the practices of active learning in our college. The first workshop had 7 participants and was conducted in March 2016. It was dedicated to the development of the design pattern model. The second workshop was conducted in May 2016 with 16 participants and the third one was in May 2017 with 19 participants. During these sessions, we collected and categorized the many problems faced by faculty as they adopted active learning pedagogical techniques in their teaching practice. We also collected and developed solutions to these problems. In some cases there were different solutions based on collaboration with slight differences for a common problem. In order to keep the patterns as simple as possible and also avoid having multiple patterns that address the same problem we added more dimensions to the solution component of the model. Hosting multiple workshops allowed us to evolve our object-based model through several iterations based on the needs of instructors that we identified in each workshop..

The next section presents the object-based model using one of our developed active learning patterns. To highlight its differences with the existing pattern formats, we compare it with a narrative pattern that address a similar problem of social construction of knowledge in lecture setting.

IV. OBJECT-BASED ACTIVE LEARNING PATTERN AND NARRATIVE LECTURE-BASED PATTERN

In order to evaluate the efficacy and flexibility of our object-based model, we present one of our developed active learning design patterns in the object-model format. This generic active learning pattern is shown in Fig. 3. The pattern addreses the problem that students need applied practice with course concepts to go beyond a theoretical understanding that they develop during lecture or during prep-work. This pattern presents a solution based on social interactions, teamwork and social construction of knowledge. Simultaneously, we present one of the lecture-based patterns from the literature (presented in Alexander's format) addressing the same problem in narrative format for fair comparison.

Pattern name: Learning activity in-class

Metadata

Pattern focus: {Learning, content delivery}
AL 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 which are performed in small groups that require the knowledge in the preparation work to complete the activity.
Rationale: Students to 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 social pressure of working in teams.
Pitfalls: 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.

Pattern Aplication

Course level {Any}
Semester: {Any}
Related CS courses: {Any}

Fig. 3. Object-based model of 'Learning activity in-class' pattern

Example of 'Learning activity in-class' pattern

Implementation Example: *Ask students to answer in-class lightweight quiz questions after consulting with others in their team.*
Formation: {Random}
Size: {medium (3-4)}
Composition: {Random}
Duration: {Entire class}
Individual grade: {Grade for individual contribution}
Contribution to final grade: {Small}
Activity progression: {Group to Individual}
Roles: {equal}

Fig. 4. Example of 'Learnng activity in-class' pattern

As shown in Fig. 3, the concise model clearly addresses students' collaboration and engagement issues. The attributes of 'meta-data' component provide higher level infromation about the pattern. Since this pattern addresses the collaboration issue between students (as read in meta-data), the teamwork attributes relate to this solution. By setting different values for teamwork attributes multiple examples can be generated for a single pattern. Fig. 4 shows a sample implementation of 'Learning activity in-class' by assigning values to teamwork attributes.

The lecture-based pattern that we have identified for comparison purpose is named 'Student miners' (AKA, social knowledge construction) [14]. The core of the problem this pattern addresses is not to "..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."[14]. This pattern format is an adapted version of the Alexandrian pattern format [10]. It contains four sections. The first section of the pattern consists of a brief description of the context, which is followed by three diamonds. In the second section, the problem and the forces are described that is followed by another three diamonds. The third section has the core of the solution in bold, the solution explains in detail the positive and negative consequences of implementing the pattern and explains possible implementations. The final section of the pattern is an examples of actual implementation that is written in italics[14].

Fig. 5 shows the narrative design pattern model in which the pattern is reprsented. To save space an overview is displayed with part of the pattern is magnified to show an example of what is written.

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 [illegible]

"….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 a new field of theory to students. Therefore, this pattern is best used with students that are not in their first year of studies …"

[illegible] answers already include the correct one, ask for more answers. Start a discussion about the differences between these answers are 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...).

Fig.5. 'Student Miners' pattern (AKA Social knowledge construction) [14]

The ‘Student miners’ pattern also offers a detailed solution and examples of how students can collaboratively work on a given problem. The other patterns that can be applied in relation to this pattern are mentioned in the form of keywords. Although the focus of this pattern is the collaboration between students, it does not offer any solution or insight about how the collaboration should be achieved or which teamwork attributes need to be considered. In the following section we discuss the observed differences and the comparison result in more depth.

A. Pattern Format Comparison Analysis

We have presented two different pattern models. One is a narrative pattern model that was previously pubished. The narrative pattern describes a lecture-based pedagogical technique. The other pattern that we presented is represented using an object-based model which captures an active learning pedagogical technique. The object-based design pattern model has more structure and both the problem and solution are more concise. By simplifying the text description for the problem and solution and creating more structure, we obviate the dimensions that are relevant for our context.

Before we are able to apply a design patter, we need to be able to search and find the corresponding pattern from a repository. As the number of patterns increases, it becomes harder to find relevant patterns that address a given problem. In this case, having an object model may help. Dimensions of the object model serve as a search criteria through which designers can narrow their search. Concise problem and solution pairing makes it easy to quickly find the problem that is being searched for and evaluate whether the provided solution would fit the designers need. The object model makes both of these tasks easier by pulling information out of the problem and solution and making the extracted information available for filtering.

The extracted information can be seen as the domain specific aspects which may differ from active learning patterns, to lecture-based patterns. These dimensions ensure that important information is presented clearly. For instance, the ‘student miner’ pattern [14] is about collaborative knowledge construction but it does not explicitly bring attention to this collaboration aspect. Teamwork attributes in the object-based model are a way to highlight these aspects. Similar to programming context, the core of object model, which includes problem, solution, pitfalls, and rationale, can be seen as an abstract class. It can be extended with different components and attributes to be applied within a specific domain. These attributes are like “strongly typed” variables which ensures that they can be reliably used for tasks such as search. In narrative design patterns, this information is often implied through the problem and solution text. These two variables can be seen as “weakly typed” variables because they don’t enforce consistent representation from pattern to pattern.

In narrative design pattern models the solution is very specific about how the pattern relates to other patterns. It doesn’t always limit these relationships to pitfalls but also describes similar patterns and originating patterns which serve as a hierarchy. The solution also provides rationale for the solution. In this way, the actual implementation can be obscured by this additional information and designers would need to look at these other patterns as well to understand the context. This may lead designers to be overly specific about how the patterns are implemented and it requires them to do a lot of additional reading and work.

In the object-based model, the solution is concise because the information is distributed throughout components of the object model and their attributes. The attributes and their values provide guidance and suggestions about how to implement the pattern, but the solution itself is very generalized which leaves it open to interpretation. This balance between specificity and flexibility is another benefit provided by the object-based model.

Another advantage of featured attributes and modularity of the object-based model is that it provides an opportunity to develop more consistent patterns in terms of their structure and their component in a given domain.

We also observed differences in the perspectives and context of the solutions that these two patterns offer (‘learning activity in-class’ and ‘student miner’). For example, the ‘Student miner’ pattern suggests that it be applied only after the first year. The rationale is that in in the first year students do not have enough prior knowledge to work on problems in class. Our active learning accounts for this problem as a pitfall. It suggests that prep-work or short-lectures might provide students with this information as needed. In this way, patterns can be applied at any level provided that the pitfalls are considered and addressed with additional patterns.

In summary we can see that object-based design pattern model can be indexed and searched because of their modular structure and defined attributes. The concise representation of the problem and solution and the attributes together means that there is a good balance between specificity and flexibility. Therefore, the object-based pattern model can be easily adapted to specific contexts such as active learning where dimensions such as team size, which are the social opportunities afforded by active learning, can be highlighted. Finally, by presenting pitfalls which link to other patterns a holistical learning environment can be created by accounting for side-effects that occur when implementing a pattern.

In the following section our method for representing the relation of the patterns and suggested sequential organization is presented in the form of concept map. We compare how using concept map helps designers navigate through the pattern space and its advantages over just mentioning the related patterns in the body of the pattern in narrative format.

B. Relational Representations of Patterns

It is challenging to evaluate patterns individually because they can have side-effects or because they do not fully address all the problems that are encountered in classroom settings. According to Alexander [12] assembling the patterns together gives more value to them and representing their relationship in a given domain is called the pattern language. In education, this pattern language is important because the needs of students are varied and the needs of each classroom vary widely depending on content, instructor’s preferences, and the physical layout of the classroom.

In this study, we apply concept map as a tool to visualize the relationship between developed object-based patterns. In Fig. 6, a concept map as an object-based pattern language represents the relationships between patterns as directional from problem-solution pair (marked by patetrn name) to pitfall of the same pattern.

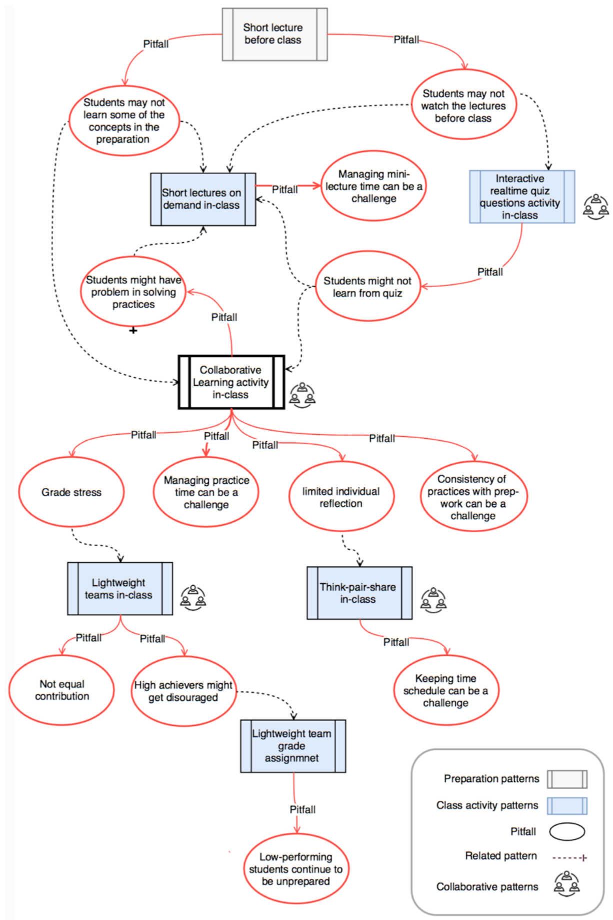

Fig. 6. Relational representation of active learning patterns as a concept map

In this concept map, the pitfall(s) of each pattern leads to an existing pattern that addresses that pitfall as a problem. In order to illustrate navigating through the concept map let's consider a preparation pattern such as 'short lecture before class' as an example. This video lecture prep work that occurs before class has a number of potential problems associated with it such as; students don't learn the material the first time that they interact with it, or they only passively engage with the material and aren't fully prepared to do in-class activities. Each of these pitfalls lead to two other patterns that address these problems. This highlights the importance of using multiple patterns together rather than choosing a single pattern that tries to mitigate every pedagogical problem. This system allows keeping the body of the patterns concise while helping easier navigation and exploration of design space. The links between patterns are not prescribed and absolute, they serve as suggested pathways through the design space. However the designers can adopt and combine patterns based on their own prefrences in any context. This self-descriptive pattern space gives flexibly to designers to choose multiple patterns that work well together.

In the narrative pattern model (as an example the 'Student miners' pattern [14]), related patterns are mentioned as keywords with capital letters directly in the body of the pattern. In this format, the relationships between patterns are sometime elaborated and some other times are implied and need to be inferred by the reader. Moreover, the names of the related patterns are not always descriptive and it makes the relation interpretation even more challenging. In the object-based pattern language however, all relationships between patterns are directional and are described in the pitfalls section. This supports the idea that 'pitfall' is an important attribute of any pattern.

In the narrative pattern format the types of relationships between patterns are defined by bolded keywords that are integrated into the narration. The reader needs to identify the type of relationship between each pattern by reading the pattern narrative. This can take a significant amount of time for the reader. We tried to identify the types of relationships mentioned in the solution of the "student miners" pattern [14]. We categorized them in four types of: Originating patterns, similar patterns, course specific patterns, and related patterns. This lack of a uniform relationship and the diversity in the types of relations makes the pattern language less consistent and more challenging to be interpreted by potential users. In the object-based model we introduce the idea of attributes which eliminates the need for similar patterns or course specific patterns in the pattern language. In this model, these varied needs are achieved by developing the examples (Fig. 4) of the abstract pattern (Fig. 3) which have different values assigned to the 'pattern core' and 'implementation' attributes. This helps to minimize the number of developed patterns and avoid redundancy. Therefore, in the resulting object-based pattern language we have a hierarchy of problem-solution pairs that generate pitfalls and the pitfalls are directed to other patterns as possible solutions.

Active learning and collaboration are often coupled with flipped classrooms. In this way, flipped classrooms provide students with an opportunity to become familiar with the materials at home, get practice with it in class, and then extend their understanding after class with assignments. This complexity means that implementing one signle patterns would not likely sufficient to create a successful collaborative learning environment. Instead, multiple patterns could be combined to deliver content consistently throughout the active learning cycle. This aspect highlights the importance of a usable and comprehensive pattern language that can be applied by designers.

In summary, the object-based pattern language or the concept map representation describes why a sequence of patterns can be combined and applied together. Narrative lecture-based patterns on the other hand suggest several patterns that are related but there is not enough clarity on the nature of relationship. As the number of patterns grows in the narrative format it would be more difficult for the designers to compile the pattern language and choose a set of related patterns. This implies that the object-based pattern language has less complexity compared to narrative pattern language.

V. DISCUSSION

We have compared a narrative lecture-based design pattern model to our object-based design pattern model. The two models were created for different purposes and therefore contain some differences in terms of what they afford to instructors and course designers. These differences also help to differentiate active learning and lecture-based classrooms.

The differences between object-based and narrative lecture-based design pattern models affect the usability of the resulting patterns. We have observed that the structure of a design pattern can have an impact on the perceived affordances. Object-based representations have pitfalls and attributes which afford search, establish meaningful relationships between patterns, and help to minimize the text in the problem and solution. Attributes and values provide information at a glance that can be easily searched. This ability to search in narrative formats is limited to the words used in the narative. Object-based model makes the patterns more readable and the instructor or course designer can quickly understand the problem and solution. Attributes also cue designers to think about specific aspects of design such as teamwork. Designers are cued to not only think about teamwork at a high-level but also think about the low-level implementation details. Pitfalls provide a sensible mapping between patterns. Pitfalls are the way to relate side-effects of implementing a pattern to other patterns which help mitigate those side-effects. This approach creates a constellation of patterns that are each related to each other. These constellations can be visualized as concept maps. In the narrative patterns, all of this information needs to be encoded into the problem or solution which reduces their readability.

One other main difference that we've observed is that in the lecture-based patterns that we reviewed, the relationships between patterns were denoted by writing the pattern's name in all capital letters. The type of relationship needed to be inferred from the context. These relationships help to show related patterns, but they require the designer to read the related pattern to understand why and how related patterns pertain to the current pattern. These relationships pointed to the provenance of patterns as well as patterns that were most related. These patterns often solved a similar pattern in a slightly different way. The designer might read many different patterns, look up related patterns, and choose the pattern that best addressed their specific problem.

Active learning classrooms require a constellation of patterns which are chosen in order to coalesce to form holistic learning environment. In this way, active learning is as much about what happens before and after class as it is about what happens during class. For example, some students have trouble learning the material the first time that they encounter it and they may need to be exposed to the information a little on their own before class via prep-work. But prep-work introduces new problems that must also be accounted for with more patterns. In this case, it is necessary to create a holistic experience for students in order to successfully create classrooms where students are able to learn both individually and collaboratively to develop both their declarative and procedural understanding of the course material at their own pace. Based on our comparison, we've seen that object-based design pattern models provide unique affordance that may make them more usable than narrative design pattern models. Object-based design pattern models are better able to capture the many complex aspects of active learning which can make adopting active learning so difficult for instructors who do not have experience teaching in non-lecture formats. Therefore, our results suggest that when designing materials for effective active learning, an object-based model may be more helpful than narrative models. Furthermore, active learning isn't about specific activities but instead, it is about creating classroom environments that support opportunities for activities at multiple points in time, often both inside and outside of the class.

One of the main goals of applying design patterns is to facilitate the process of sharing teaching practices in a structured format in the form of problem-solution pairs. They do not prescribe solutions, as the process of design and development of design patterns is an incremental process, patterns evolve over the time in different contexts. The object based model of design pattern makes this evolutionary process easier, since diverse solutions can emerge and be derived from the generic and abstract solution based on instructors' skill and experience.

Another advantage of applying pedagogical design patterns is to provide a meta-level view on teaching practices. During several workshops we had with faculty to develop design patterns, we noticed some instructors intutively identify the problems they face in the cousre and have well stablished methods and solutions to those challenges, but it was not easy to formalize them as a design pattern in the form of problem solution pairs. Design patterns help instructors to have a metalevel view of their own practices. We believe this modular formalization of teaching practices help instructors (especially the less experienced ones) in adopting a meta-cognitive process for course design.

VI. CONCLUSION

In this paper we present our pedagogical design pattern model which captures the collaborative aspects of active learning. We reviewed existing published design patterns and design pattern models that have been employed in CS education. We observed that these pattern models, which were developed based on Alexander's pattern format, were presented in narrative form and emphasized mostly lecture-based pedagogies and methods.

We proposed an object-based model to represent design patterns that also uses Alexander's format in its core, but it also includes extended components with attributes to acheive a modular pattern structure. The modularity of this object-based model helps designers search and index the patterns. The meta-data component of the object-based model provides higher level information about the pattern. This feature gives flexibility to designers to decide to apply the pattern without having to read through the whole pattern. The modular structure of the object-based model and its attributes prevents redundancy and developing similar patterns with minor differences. The object-based model supports a fundamental idea of design patterns which suggests the patterns should be simple and elegant [13]. Pedagogical design patterns should be able to be implemented many times without having to implement the same solution twice [11]. The object-based model supports having different solutions that address the same problem by adopting different values for the attributes of the components. Another contribution of this work is the relational representation of the patterns. Because instructors face multiple problems when designing their courses, patterns are most valuable when combined. Therefore, we used the idea of concept map to represent the relationship between patterns.

In summary, the contributions of this paper are:

1) Presenting an object-based pedagogical design pattern model with attributes that has the flexibility to be applied in any pedagogical setting.

2) Providing an example of active learning pattern in object-based model which address the problem of social construction of knowledge.

3) Comparing a pattern in the object-based model with a narrative pattern that addresses the same problem, and highlighting the differences.

4) Proposing an object-based pattern language by applying concept map as a tool for relational representation of the patterns.

5) Comparing two types of pattern languages: a) our developed concept map and b) description of relations in narrative pattern format, and discussing the differences.

In future work, we will evaluate the relevance of a concept map of patterns. We will also evaluate the pattern model in different CS courses to see how this object-based model encourages instructors to apply successful active learning techniques.

REFERENCES

[1] Köppe, C., & Portier, M. (2014, July). Lecture design patterns: improving the beginning of a lecture. In Proceedings of the 19th European Conference on Pattern Languages of Programs (p. 16). ACM.

[2] Prince, M. (2004). Does active learning work? A review of the research. Journal of engineering education, 93(3), 223-231.

[3] Bonwell, C. C., & Eison, J. A. (1991). Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports. ERIC Clearinghouse on Higher Education, The George Washington University, One Dupont Circle, Suite 630, Washington, DC 20036-1183.

[4] Authors order be defined. (2018) A Pedagogical Design Pattern Model for Collaborative Active Learning. Computer Science Education journal, (manuscript ready for submission)

[5] I 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.

[6] Millis, B. J., & Cottell Jr, P. G. (1997). Cooperative Learning for Higher Education Faculty. Series on Higher Education. Oryx Press, PO Box 33889, Phoenix, AZ 85067-3889.

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

[8] Narayanan, N. H., Hundhausen, C., Hendrix, D., & Crosby, M. (2012, February). Transforming the CS [13classroom with studio-based learning. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 165-166). ACM.

[9] Haberman, B. (2006). Pedagogical patterns: A means for communication within the CS teaching community of practice. Computer Science Education, 16(2), 87-103.

[10] Alexander, C. (1977). A pattern language: towns, buildings, construction. Oxford university press.

[11] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). A pattern language: Towns, buildings, construction (center for environmental structure).

[12] Goodyear, P., & Yang, D. F. (2009). Patterns and pattern languages in educational design. In Handbook of research on learning design and learning objects: Issues, applications, and technologies (pp. 167-187). IGI Global.

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

[14] Köppe, C., & Schalken-Pinkster, J. (2013, October). Lecture design patterns: improving interactivity. In Proceedings of the 20th Conference on Pattern Languages of Programs (p. 23). The Hillside Group.

[15] Michaelsen, L. K., & Sweet, M. (2008). The essential elements of team‐based learning. New directions for teaching and learning, 2008(116), 7-27.

[16] 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.

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

[18] Dehbozorgi, N. (2017, August). Active Learning Design Patterns for CS Education. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*(pp. 291-292). ACM.