Using Spectrums and Dependency Graphs to Model Progressions from Introductory to Capstone Courses

Stephen MacNeil, Mohsen Dorodchi, Nasrin Dehbozorgi
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC, USA
{smacnei2, mohsen.dorodchi, ndehbozo}@uncc.edu

***Abstract*— In industry, professionals often work with a variety of stakeholders and collaborators from multiple disciplines. This ability to work collaboratively can be as important to a project's success as their technical skills. Traditionally in STEM education, these collaborative skills are developed in a capstone course which mimics an industry experience. These experiences are invaluable in preparing students for the collaborative real-world nature of industry; however, these experiences can also be very stressful for students in dysfunctional teams with members who haven't developed necessary social, technical or teamwork skills. Although students may be exposed to some team-based activities in previous courses, it is not clear that this piecemeal exposure teaches students to work in teams effectively. Flipped classroom and active learning attempt to fill this gap by exposing students to peer learning earlier in the curriculum. However, these techniques are peppered throughout the curriculum and may not target all the skills necessary for teamwork.

Design patterns in education formalize pedagogical approaches. But, applying design patterns without an intended progression or overarching goal may not lead students to successfully adopt these skills. Design patterns have the potential to scaffold students' development throughout the curriculum, but only if staged effectively and systematically. In this paper, we propose Spectrums and Dependency Graphs to ensure that students are prepared for each new design pattern as they experience it. Spectrums can plot design patterns along a continuum between introductory and capstone courses. Dependency graphs recursively specify patterns that prepare students for subsequent patterns. Each pattern will contain prerequisite skills or experiences that students have demonstrated in a previous pattern. In this way, students are systematically progressed from introductory to capstone courses. Through these two models, we attempt to get a better overview of the curriculum and create progressions through that curriculum that ensure students are prepared at each level, building on previous skills.

Keywords*— *design patterns; active learning; team-based learning; capstone; curriculum design; dependency graphs; spectrums; CS1; CS2

I. Introduction

Peer learning is becoming more ubiquitous throughout the curriculum. Traditionally associated with project-based learning and capstone courses in engineering, peer learning in teams has become a common way to support students in active learning classrooms. During class time, there is only one instructor to help the entire class and so grouping the students is a natural way to help students in a more scalable way. Peer learning in the context of introductory courses and capstone courses have justifiably received a lot of attention from many research communities. A good introductory experience can help students make friends and develop a support structure, which has been consistently linked to retention [16]. Good introductory experiences help students develop a foundation that will serve them through their academic career, and it can help them to develop self-confidence and self-efficacy [10]. However, a negative first experience can reinforce self-doubt and make them reconsider their decision to major [13]. In the same way, a positive experience at the capstone level can provide students with a strong portfolio and the necessary communication skills to obtain a good job that serves as a spring board for their careers. A negative experience at the capstone level may push students to re-evaluate their career goals and potentially choose a related but different field, such as sales, techincal recruiting, or quality assurance.

Taking a more holistic view of education is important because the experiences that happen after the introductory courses but before the capstone courses are integral in preparing students for upper-level courses such as capstones. As our research community continues to delve into these issues to give students a strong technical and social foundation as well as a good start in their careers, we should also consider the things that happen in between these two experiences. A large portion of students' learning experiences come from courses that aren't introductory or capstone. Therefore, it is reasonable to start considering the courses and experiences between these two poles as a continuum where the goals at each extreme, introductory and capstone courses, are blended. By taking a systematic approach to considering these intermediate courses we can start to design progressions that help scaffold skills to ensure that students are prepared when they begin their capstone courses and careers.

Design patterns have been used in architecture [11] and in education [12] as a way to formalize possible solution for a well-known problem. In education, these may capture a common problem like "team formation is a challenge for instructors" and provide possible solutions, such as "forming teams based on surveys, or randomly or students self-select their peers, etc.". These patterns typically exist as they are, but may be linked to other related patterns. To our knowledge, no

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

work has considered these patterns as existing along a spectrum or as having pre-requisites. In education, these patterns may address problems that only exist in introductory classes but not in capstone and vice-versus.

In this paper, we describe some of the differences between the purposes of introductory and capstone courses. We illustrate that a spectrum exists between these two poles and describe some pathways that can be plotted along this spectrum. We provide a case-study that illustrates how spectrums can be helpful to situate patterns along the curriculum and we describe how they can be combined with dependency graphs to create well-progressed learning experiences for students.

II. Background

Team-based activities can take place throughout the curriculum. They begin in introductory courses where students work collaboratively to solve closed-ended well-defined problems. In the capstone courses, they work on ill-defined problems with little intervention from the faculty. They are more responsible for their learning and for making their way through course material. But how do we prepare students for this experience? In the following three sections, we outline what team-based activities look like in both introductory and capstone courses. Finally, we describe the necessary attributes a capstone student should possess as grounded in literature.

A. Team-based Activities in Introductory Courses

Active learning and team-based learning are two common ways to support students in early courses. They force students to negotiate understanding with their peers and integrate multiple perspectives. These activities also have been helpful for improving student motivation and retention [3,4,5,6,7,8]. Team-based learning, as a part of active learning, also helps improve student's technical, social and leadership skills [9]. When compared with lecture, peer instruction [21] was preferred by students and perceived as being more enjoyable [20]. In peer instruction, students answer questions collaboratively and share insights with eachother. A study with 6000 students indicated that interactive engagement strategies supported students' problem-solving skills and their conceptualization and understanding of the concept [4].

The focus of active learning in introductory courses is on social interactions, social support, friendship, along with learning the concepts. However, advanced teamwork skills cannot be practiced due to students' varied background levels and the load of the technical material that they are learning.

B. Team-based Learning in Capstone Course

Capstone courses were traditionally seen as a way for students to apply all of their knowledge in a project-based learning experience that prepared them for industry. Students can use these projects as evidence that they are well-rounded and know how to work effectively with others. The hallmarks of a capstone course are that students engage as a team in task identification, design specification, project prototype development, scheduling, and prototype testing [15]. Based on the results of a questionnaire of capstone instructors by Todd et al., 45% of instructors reported that the duration of these capstone courses were one semester and an additional 36% reported that they taught a two-semester capstone course [14]. Capstone courses can also include industry partners to create a pipeline for employment and provide real-world experience.

The focus of these capstone-level courses is on performance and demonstrating skills as evidenced by the fact that the grade often consists of project prototype, project documentation, and a final presentation. This focus is very different from the introductory courses that have focused on learning specific technical and social skills in a supportive environment.

C. Preparedness for Capstone

In the introductory courses, the focus is on student engagement and having students develop their skills with the help of an instructor or teaching assistant (TA). These are stress-free environments that foster positive interdependence between students in teams [17]. Significant scaffolding is necessary and can be achieved through timed and structured activities [18]. This learning environment is very different from a capstone experience where students are expected develop an artifact and they are responsible for scoping the problem, budgeting their time, and working effectively in teams. Research often focuses on these two contrasting environments in isolation; however, to our knowledge few papers have explored the progression that is necessary to guide students from very structured, scaffolded learning environment to develop the skills necessary to succeed at the capstone level, where they are expected to be autonomous and demonstrate individual accountability [19].

III. Progressing from Introductory to Capstone

In our work with design patterns, we identified that a gap exists in how patterns are combined and stacked to ensure that students progress through the program and develop the skills necessary to be successful in capstone courses and beyond. Students may be asked to discuss a topic in groups for one course, they may assume a timekeeper role to ensure the group stays on track in a different course, and they may have to decompose a problem into solvable sub-problems in a third course. However, these piecemeal experiences with team-work do not guarantee that students will be well prepared to assume roles successfuly on teams with short deadlines on open-ended problems that require significant autonomy on the part of the students. The highly scaffolded and structured environment that students encounter in their first years should evolve systematically to ensure that students are obtaining all of the necessary attributes or skills expected in the capstone.

A. Spectrums spanning the Curriculum

To effectively progress students from introductory to capstone courses it is necessary to consider what types of patterns they are likely to encounter and at what points in their academic careers. Based on a survey of instructors in our department and a review of capstone research we have established a set of design patterns for per learning. In reviewing these patterns, they typically fell into two categories: activity-based teams which represent group work in the introductory courses and project-based teams which represent the capstone level team-based learning experiences. Looking at them as two poles on a continuum, we noticed a gap between these two extremes. There were few patterns that applied to both conditions and even fewer that existed between these two poles. We began to consider whether there could be a set of patterns that existed on the continuum between these poles. This thought exercise, led to the concept of spectrums. Spectrums are continuums between introductory and capstone courses. Design patterns can be plotted along the continuum as shown in Figure 1. Spectrums help us step back from thinking about learning experiences as being black and white and think about what exists in between.

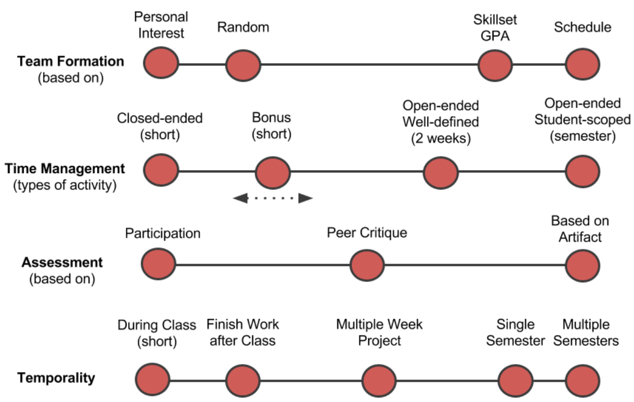

Fig 1. Four example Spectrums from Introductory to Capstone.

If we consider Team Formation as a spectrum, students in the introductory courses may be formed randomly to avoid homophily or based on complementary personalities. However, at the capstone level these concerns may not be as relevant. Students may instead be grouped based on their skillset so that each team has an expert in each sub-domain, such as databases, web-development, testing, etc. Or they may be grouped based on their schedule to accommodate work schedules so that students can meet outside of class time.

There are a range of patterns that don't neatly fit into either of these two poles. They may contain equal aspects of each extreme or they may be more related to the capstone experience or the introductory experience. For instance, we can consider Time Management as a spectrum. Students start in introductory classes with a short activity that has very clear step-by-step instructions with expected time requirements. Students aren't required to budget their time because it is done for them. The instructor may even announce when they should be moving on to the next step. A little further along the spectrum, they may not receive the time required for each step and may receive a bonus if they budget their time well and make it to the final step in the allotted class time. Further still, we can imagine a well-defined group project that has well-specified outcomes but the steps aren't described explicitly. Students are given a week or more to complete the project and they must deconstruct the problem and manage the time themselves. Finally, students would be ready for a capstone experience where they are expected to identify and scope the problem themselves. In this case, each pattern builds on the previous; however, this is not required. The spectrum doesn't describe an explicit dependency it just describes at what approximate level a student would expect to encounter a pattern. In the case of Time Management, the spectrum describes how much autonomy students have in terms of time management. And there are a few patterns that exist along the spectrum where students have increasing amounts of autonomy as they move through the curriculum.

B. *Dependency Graphs and Matrices*

Dependency graphs make explicit the attributes and skills a student should possess in order to be considered prepared to participate in a learning activity described in a pattern. Pre-requisites describe these necessary attributes and skills that students should possess. In the previous example, the final pattern required that students can manage time, deconstruct problems into their constituent parts, and negotiate a project timeline with team-members while considering schedule conflicts. They should also have experience working on longitudinal projects that take place outside the classroom.

Making these dependencies explicit affords an opportunity to graph these dependencies. Given a specific pattern with a set of prerequisites or attributes, there would be a set of patterns in which students have demonstrated their ability to meet these requirements. Those patterns are plotted as inputs to the given pattern. If those patterns have prerequisites, they would have a set of patterns that meets those prerequisites recursively until all requirements are met for all patterns.

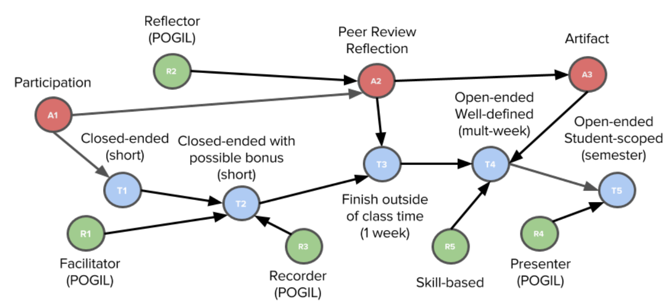

Fig 2. A Dependency Graph for a Capstone time-management pattern. Nodes from time management (blue), team roles (green), and grading focus (red) are shown as dependencies of each other. Redundant paths are not shown. For instance, participation (A1) is a dependency of nearly every task (T1-5).

As a case-study, we start by choosing a pattern to implement: the open-ended student-scoped pattern (T5). This pattern is common at the upper level courses because it has many prerequisites, such as timekeeping, problem-decomposition, and teamwork. The dependency graph for this pattern (Figure 2) describes how these prerequisites are met by specific patterns. From the chosen goal-pattern, we see that we can work our way backwards and see that to accomplish open-ended, student-scoped tasks (T5), they must have previous experience with open-ended tasks that are well-scoped by the instructor (T4). Open-ended tasks are often assessed by the resulting artifact, and students should have experience working on projects where the artifact is the primary source of assessment (A3). And since these tasks may take multiple weeks, they should have experiences with tasks that take place outside of the class (T3). Finishing work outside the classroom requires students to communicate to the instructor about whether their team-mate is contributing equally through some form of peer review (A2). For longer activities such as T3 and T4, a timekeeper or facilitator (R1) is needed to meet the deadline and budget time. Closed-ended tasks with a bonus for students who finish on time is a way to start getting students to consider timekeeping and estimating how long tasks take to complete (T2). The main form of assessment in these closed-ended tasks which appear in introductory courses are participation (A1) to encourage students to work together in a low-stress way since they haven't fully developed teaming skills yet.

Dependency graphs can also be represented as an adjacency matrix. Along each row, if the design pattern has a dependency it would be represented by a 1 and a 0 otherwise. This type of representation would be more concise, but it might be harder to illustrate dependency paths visually, such as those that exist in Figure 2 in a matrix form.

We can see that dependency graphs are useful in ensuring students develop attributes and skills that are necessary to be successful with a given pattern. Such graphs help curriculum designers create a progression through the curriculum in which students are exposed to increasing amounts of autonomy, roles, and responsibility. These dependencies ensure that patterns build on each other culminating in multidimensional students who are well-prepared at each level.

IV. Discussion and Future Work

We have presented spectrums and dependency graphs to give an overview and aid in the creation of progressions through the curriculum. Spectrums represent design patterns (such as a learning activity) along a continuum from introductory courses to capstone courses. Dependency graphs make explicit the dependencies that each design pattern has and allows us to create systematic progressions through the curriculum.

Spectrums give us an overview for which patterns exist where. Gaps can be identified along these spectrums where few patterns exist. This may prompt the creation of new patterns that bridge these gaps between the introductory and capstone courses. We have presented 4 spectrums in this paper (Figure 1), but we can imagine many other spectrums along which to place design patterns. This is novel way of viewing patterns, which are typically presented only as a list.

Dependency graphs are a way of staging patterns in such a way that students are prepared for each design pattern that they encounter it. Figure 2 shows a dependency graph where each pattern prepares students for the next pattern that they experience. Such graphs, help create systematic progressions through the curriculum that ensure students are prepared at each level in the curriculum.

We have proposed the concepts of spectrums and dependency graphs as two ways of reflecting on design patterns. We created anecdotal examples of each to illustrate the idea, but these anecdotes are not finished or completely valid. The specific order along the spectrums and the dependencies portrayed in the dependcy graph are only to illustrate the idea. They are rooted in our experiences, but specific dependencies for each pattern need to be identified and they must be rooted in theory and surveys of instructors. We have only proposed the concept of how design pattern dependencies can be modelled sequentially.

Another area of interest that is not covered in this paper is the way in which design patterns interact with each other. Combining design patterns in a single class is relatively common, and these interactions and combinations may have beneficial or detrimental effects. In this work, we don't investigate this tangential possibility, but only focus on how individual design patterns can be sequenced. It is likely that combinations of patterns could be sequenced in dependency graphs and this possibility might more closely resemble a model for the curriculum.

Based on this work, we can start to think about gaps in the curriculum and ways to sequence patterns so that students are prepared as they encounter new educational design patterns. Without such models available, design patterns may not systematically lead students to progress through the curriculum effectively. Identifying gaps may be useful for ideating new patterns that more effectively bridge the gaps between introductory and capstone courses. This would likely result in a smoother progression through the curriculum for students.

V. Conclusion

In this paper, we have proposed spectrums and dependency graphs. Spectrums give a sense for where educational design patterns exist throughout the curriculum. Dependency graphs ensure that students progress through the curriculum in such a way that they are prepared for each new learning environment as they encounter it. Future work is needed to populate these spectrums and dependency graphs with patterns. These patterns need to have dependencies that are rooted both in theory and in the experiences of instructors. After populating these spectrums and graphs, instructional designers may be able to ideate new design patterns that better cover the space and create systematic progressions.

References

[1] J. Pirker, M. Riffnaller-Schiefer, and C. Gütl, "Motivational active learning: Engaging university students in computer science education," inProceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, ser. ITiCSE '14. New York, NY, USA: ACM, 2014, pp. 297–302. [Online]. Available: http://doi.acm.org/10.1145/2591708.2591750

[2] R. Sowell, Y. Chen, J. Buhler, S. A. Goldman, C. Grimm, and K. J. Goldman, "Experiences with active learning in cs 3," Journal of Computing Sciences in Colleges, vol. 25, no. 5, pp. 173–179, 2010.

[3] J. D. Bransford, A. L. Brown, R. R. Cocking et al., How People Learn: Brain, Mind, Experience, and School. National Academies Press, 2000.

[4] R. R. Hake, "Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," American Journal of Physics, vol. 66, no. 1, pp. 64–74, 1998.

[5] P.Laws, D.Sokoloff, and R.Thornton,"Promoting active learning using the results of physics education research," vol. 13, pp. 14–19, 1999.

[6] M. Prince, "Does active learning work? a review of the research," Journal of Engineering Education, vol. 93, no. 3, pp. 223–231, 2004. [Online]. Available: http://dx.doi.org/10.1002/j.21689830.2004.tb00809.x

[7] E. F. Redish, J. M. Saul, and R. N. Steinberg, "On the effectiveness of active-engagement microcomputer-based laboratories," American Journal of Physics, vol. 65, no. 1, pp. 45–54, 1997. [Online]. Available: http://dx.doi.org/10.1119/1.18498

[8] R. K. Sawyer, The Cambridge handbook of the learning sciences. Cambridge University Press, 2005.

[9] D. W. Johnson, R. T. Johnson, and K. A. Smith, "Cooperative learning returns to college what evidence is there that it works?" Change: The Magazine of Higher Learning, vol. 30, no. 4, pp. 26–35, 1998. [Online]. Available: http://dx.doi.org/10.1080/00091389809602629

[10] M. B. Rosson, J. M. Carroll, and H. Sinha, "Orientation of undergraduates toward careers in the computer and information sciences: Gender, self-efficacy, and social support," Transactions of Computing Education, vol. 11, no. 3, pp. 14:1–14:23, October 2011. [Online]. Available:http://doi.acm.org/10.1145/2037276.2037278

[11] C. Alexander, S. Ishikawa, and M. Silverstein, A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure), O. U. Press, Ed. Oxford University Press, 1977.

[12] P. Goodyear, "Educational design and networked learning: Patterns, pattern languages and design practice," Australasian Journal of Educational Technology, vol. 21, no. 1, 2005.

[13] D. McIlroy, B. Bunting, K. Tierney, and M. Gordon, "The relation of gender and background experience to self-reported computing anxieties and cognitions," Computers in Human Behavior, vol. 17, no. 1, pp. 21–33, 2001. [Online]. Available: http://doi.org/10.1016/S0747-5632(00)00037-6

[14] R. H. Todd, S. P. Magleby, C. D. Sorensen, B. R. Swan, and D. K. Anthony, "A survey of capstone engineering courses in North America," Journal of Engineering Education, vol. 84, pp. 165–174, 1995.

[15] J. G. Flowers, "Improving the capstone project experience: A case study in software engineering," in Proceedings of the 46th Annual Southeast Regional Conference on Software Engineering, ser. ACM-SE 46. New York, NY, USA: ACM, 2008, pp. 237–242. [Online]. Available:http://doi.acm.org/10.1145/1593105.1593167

[16] V. Tinto, Leaving college: Rethinking the causes and cures of student attrition. Chicago, IL: University of Chicago Press, 1987.

[17] D. W. Johnson and R. T. Johnson, "Positive interdependence: Key to effective cooperation," Interaction in cooperative groups: the theoretical anatomy of group learning, pp. 174–199, 1992.

[18] H. Hakimzadeh, R. Adaikkalavan, and R. Batzinger, "Successful implementation of an active learning laboratory in computer science," inProceedings of the 39th Annual ACM SIGUCCS Conference on User Services. SIGUCCS'11. New York, NY, USA:ACM,2011, pp.83– 86. [Online]. Available: http://doi.acm.org/10.1145/2070364.2070386

[19] D. W. Johnson, R. T. Johnson, and K. A. Smith, Active learning: Cooperation in the college classroom. ERIC, 1998.

[20] S.MacNeil, C.Latulipe, B.Long,andA.Yadav," Exploring lightweight teams in a distributed learning environment," in Proceedings of the 47th ACM Technical Symposium on Computing Science Education, ser. SIGCSE '16. New York, NY, USA: ACM, 2016, pp. 193–198. [Online]. Available:http://doi.acm.org/10.1145/2839509.2844577

[21] C. H. Crouch and E. Mazur, "Peer Instruction: Ten years of experience and results," American Journal of Physics, vol. 69, no. 9, pp. 970–977, 2001. [Online]. Available: http://dx.doi.org/10.1119/1.1374249