Drawing Insight from Student Perceptions of Reflective Design Learning

Thomas V. Wilkins and John C. Georgas
Department of Electrical Engineering and Computer Science
Northern Arizona University
Flagstaff, Arizona 86011–0564
{thomas.wilkins, john.georgas}@nau.edu

***Abstract*—While design and designing are core elements in computer science and software engineering, conventional curricular structures do not adequately support design learning. Current methods tend to isolate the study of design within specific subject matter and lack a strong emphasis on reflection. This paper reports on insights and lessons learned from a user study in the context of ongoing work on developing an educational intervention that better supports design learning with a particular emphasis on learner-driven reflection. Insights drawn from this study relate to general aspects of design learning, such as the importance of collaborative reflection and the impact of learner perceptions regarding their abilities, as well as to specific improvements to our approach.**

***Index Terms*—software engineering education, software design, design learning, reflection, structured reflection.**

I. Introduction

Design is a fundamental activity in software engineering, with design decisions made throughout the entire development lifecycle and impacting artifacts that span multiple layers of abstraction, from code-level decisions over program control flow to high-level concerns such as object-oriented hierarchies and software architecture compositions. Given its ubiquitous nature, design significantly impacts software quality, particularly non-functional qualities such as maintainability and extensibility, which is even recognized [1] by proponents of agile methodologies that often deemphasize the importance of up front design. These impacts of design naturally imply that design learning should be a critical element of software engineering and computer science training programs.

Despite the importance and centrality of design, however, conventional curricular structures in undergraduate training programs foster design learning in only limited ways: Design learning is commonly only explicitly addressed in few curricular points and within limited subject matter contexts, which unduly decontextualizes design and deemphasizes its wide-ranging impacts. In addition, the instructional methods used often do not include a strong focus on learner reflection, which is particularly important for design learning. As a result, current practice makes it difficult to support design learning in a way that appropriately addresses common design process elements in the context of the wide-ranging and diverse spectrum of software engineering design decisions.

In order to address these challenges, we are developing an educational approach that centers on learner reflection and cohesively supporting design learning throughout a wide curricular spectrum: Modular, content-specific *design challenges* reinforce subject matter learning and give learners the opportunity to navigate an appropriately-scoped design space that exercises their design skills. These design challenges may relate to various topics within software engineering, ranging from selecting appropriate data structures to composing object-oriented systems. By using a *structured reflection* framework that addresses fundamental aspects of design, learners then reflect on their solutions to design challenges and ultimately develop *design story* narratives. These narratives explicitly capture the important design decisions they made and the alternatives they evaluated across a wide range of commonly considered design concerns. Our work is characterized by a focus on learner-driven reflection and an effort to create a balanced approach that retains a cohesive perspective on design learning while remaining broadly applicable across a range of subjects.

This paper extends previously published work [2] and reports on specific insights drawn from recent formative evaluation efforts centering on a focus group study with undergraduate computer science students, who compose an important target learner population. One important generally applicable lesson drawn from this study is the importance of ensuring a high degree of congruence between the difficulty of design problems and perceived learner competence. Another such lesson is the centrality of collaborative reflection in activities relating to design learning. We also discuss insights regarding improvements to our specific approach and associated artifacts and how these insights shape our future work plans. In addition to these core contributions, we present a discussion of our specific approach to supporting design learning as well as qualitative data drawn from our user study.

II. Background and Related Work

This section provides an overview of the foundational concepts that underpin our work in design learning, discussing design theory and design rationale as well as insights drawn from pedagogical theories on supporting effective learning.

A. Learning Theories

The foundations of our pedagogical approach are based on constructivism [3], which emphasizes the importance of

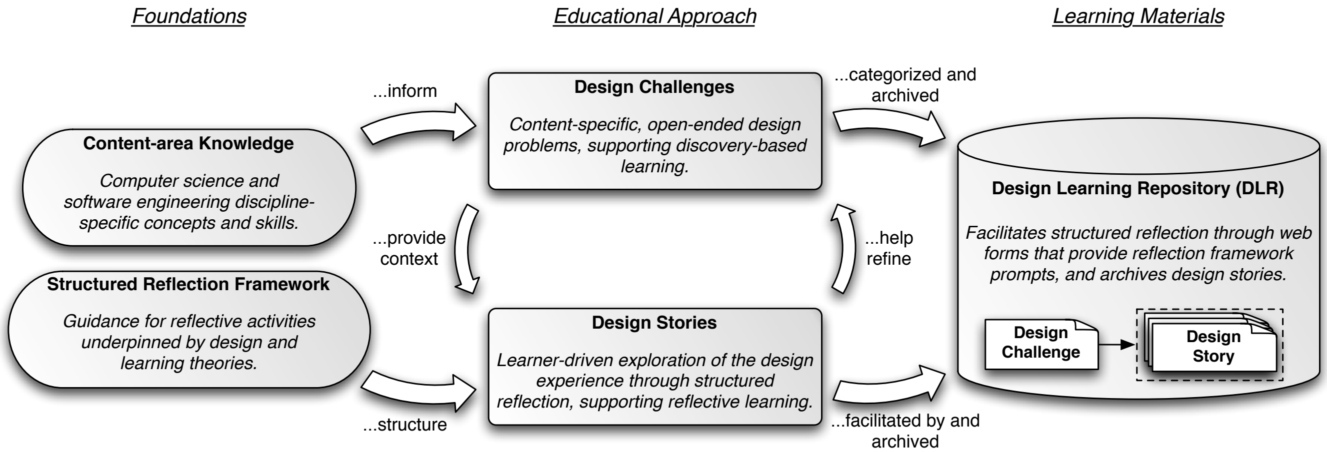

Fig. 1. An overview depicting the main processes and artifacts in our approach to supporting design learning.

creating knowledge through learner interactions. Experiential [4] and discovery-based learning [5] are important expressions of general constructivist principles and exemplify a focus on supporting learning through the interactive exploration of application-oriented problems. Our use of content-specific design challenges is predicated on this type of learning context, which also supports the development of stronger learner autonomy and problem-solving skills [6]. The centrality of reflection in our work draws from insights on the importance of reflection as a key element of design learning [7]. While reflection may take place as an open-ended process, we elect to use a structured reflection [8] approach that guides and prompts learners to consider and reflect over specific aspects of design that are commonly found and particularly relevant in the disciplinary context of computer science and software engineering. Our work also strongly incorporates insights from elaboration theory [9] by incrementally increasing the complexity of design tasks that learners engage in, beginning with creating solutions to design challenges before transitioning to the more complex task of assessing and reflecting over these solutions.

B. Design Theory

While individual areas of computer science and software engineering commonly develop design methods and associated terminology, there are important lessons to draw from discipline-independent conceptualizations of design. In general, these conceptualizations frame design as a decision-making process that considers a range of design alternatives that constrain and shape the final characteristics of a solution. Simon [10] broadly considers design as a hierarchical decision-making process that considers the constraints which limit design options and alternatives, while a similar perspective with an emphasis on predicting the future effects of design decisions is adopted by Marples [11] in the context of engineering design. In our work, we draw on these general perspectives when aiming to focus the attention of learners on common design concerns that are found across multiple software artifacts and levels of abstraction. Alexander [12] points out the importance of considering the impact of change, thinking of design as a dynamic artifact that is adapted over time to improve its quality and conformance to problem constraints. His work also emphasizes design patterns as a key medium through which design knowledge is retained, which is a perspective that strongly informs elements of our reflection framework that aim to elicit a better understanding of constraints and explicitly prompt the consideration of helpful design patterns. Work by Jones [13] considers design to first diverge from an intended solution, as designers build a better understanding of the problem space and explore the impact of alternatives, before converging toward an ultimate solution through incremental improvements. This perspective resonates with our focus on reflection as a means of fostering understanding and learner consideration of design aspects that may have been overlooked in the initial solution.

C. Design Rationale

Design rationale [14] is focused on capturing the context behind design decisions, including an explicit consideration of alternatives for each decision and the reasons why they were not selected. The attention our approach devotes to fostering an exploration of alternatives reflects the importance of design rationale in the context of software system development. Unfortunately, the widespread use of design rationale capture techniques suffers from challenges [15] relating to the high overhead they impose on designers, the obtrusiveness of associated tools, and the lack of an immediate reward for engaging in the rationale capture process. While our approach does aim to capture some measure of rationale through learner reflection, it also exhibits distinguishing features that are aimed at overcoming difficulties associated with design rationale capture: First, the context of design challenges is limited, which results in comparatively little effort when reflecting over and recording design decisions. Second, the structured reflection activity is learner driven and does not rely on automated capture tools. Finally, there is a clear reward structure for participants, as the entire activity is cast in the light of an immediately valuable learning experience.

III. Approach

Our approach is aimed at fulfilling three key goals: (a) to provide a method for cohesively supporting design learning throughout the entirety of the computer science or software engineering curriculum, which demands an increased focus on generally-applicable aspects of design that are largely decoupled from specific design artifacts and contexts; (b) to explicitly include reflection and its outcomes as core elements of the design learning process, which requires artifacts and an educational intervention design that strongly incorporates learner reflection as a key activity; and (c) to support a modular educational intervention strategy that allows educators to adopt this approach in a way that is minimally disruptive to existing curricular structures.

Central to fulfilling these goals is the use of modular, content-specific *design challenges* that allow learners to explore an open-ended design space within an appropriately-scoped context that varies depending on the subject matter used as the basis for developing design challenges. Solutions to design challenges provide learners a foundation for learning from their design decisions and exploring concepts they may have overlooked through conceptual framework-guided *structured reflection*. This reflective activity allows learners to build reflective narratives we refer to as *design stories*. The deployment, development, and archival of design stories is facilitated by an accompanying web-accessible digital archive. More details on each of these elements appear in the following sections, with a graphical overview of the approach appearing in Figure 1.

TABLE I
A REDACTED GENERAL-PURPOSE STRUCTURED REFLECTION FRAMEWORK FOR SUPPORTING DESIGN LEARNING.

Design Aspect	Structured Reflection Questions
Functional Requirements	What were the main functions needed? Which of these was the most critical and why? Did you discover unstated but necessary functions?
Non-Functional Requirements	What non-functional properties were needed? Which of these was the most critical and why? What were the non-functional property trade-offs?
Decision Points	What decisions were made for each needed function? What alternatives were considered for each decision? How did decisions limit future design options?
Design Principles	What design principles did you use in this challenge? How did each principle influence your design? Which principle did you find most helpful and why?
Design Notations	What notations did you use to capture your design? What design needs guided your choice of notation? Which notations were the most and least helpful?
Knowledge Preservation	Did your solution use known design patterns? Can other patterns be extracted from your solution? What else did you learn while solving the challenge?

A. Design Challenges

The deployment of our approach begins with educators providing a short overview of a design challenge (DC), pointing out important characteristics that may be particularly relevant to the content targeted by the posed problem, and allowing learners to solve the problem as individuals or in small groups. DCs form the foundation for modularly distributing design learning throughout a broad range of computer science and software engineering curricular structures by pairing the development of design knowledge with domain-specific learning. The subject matter for DCs is drawn from the specific learning outcomes associated with the educational context in which they are used. In an introductory programming course, for example, DCs would center on the composition of basic programming constructs into algorithms for solving simple problems, while a course on user interfaces would naturally be oriented toward design problems focused on the function and usability of a graphical user interface—an example of a DC drawn from a second-semester introductory programming course appears in Section IV-A. The content-specific and fine-grained nature of DCs allows both educators and learners to couch design learning within an immediately applicable and familiar context that is not overly demanding of in-class time, therefore providing the means for a minimally disruptive integration within existing pedagogical structures.

B. Structured Reflection

The process continues with learners engaging in learner-driven reflection, which we consider the primary means for supporting design learning. While open-ended reflection can be a valuable tool through which to improve knowledge, we see a key shortcoming in failing to provide some measure of structure: Beginning learners often do not yet have the expertise to be aware of important considerations in reflecting over their solutions to problems. As a result, they may inadvertently miss important learning opportunities.

We address this challenge by providing guidance to learners in the form of a structured reflection framework that captures generally applicable aspects of design—with significant redactions for brevity, the framework we use with learners appears in Table I. The design of the reflection framework is intended to reduce the coupling of the reflective activity with specific disciplinary areas of expertise. This provides a way for learners to maintain a cohesive perspective on common aspects of design, such as focusing on a clear understanding of requirements and the use of patterns, as they engage in the activity of designing across a wide spectrum of disciplinary contexts and topics.

C. Design Stories

As learners engage in the structured reflection process, they incrementally create what we refer to as a design story (DS): a narrative (primarily textual with informal diagrammatic annotations) that explicitly captures the results of learner reflection over their problem-solving experience. Of course, this may be followed by a re-factoring of design solutions, as reflection exposes shortcomings or ways in which solutions can be significantly improved. By preserving and archiving learner-created DSs (discussed in Section III-D), our approach also provides the foundation for extending the conventional role of learners as relatively passive participants in the learning process. Through these narratives, learners become active contributors to a body of knowledge that accumulates over time and—as educators choose to share relevant DSs with other learners—underpins the learning gains of their present and future peers.

D. Web-enabled Support

The preservation of DSs is achieved through a web-based system we call the *Design Learning Repository* (DLR), constructed using the Django framework with a ZURB Foundation front-end and a MySQL database. The system provides support for functions aimed at streamlining the tasks involved in adopting our approach: Instructors can create course shells, enroll students in specific shells (which prompts students to create a DLR user account), and assign DCs by creating their own problem descriptions or selecting from a set of challenges organized according to conventional curricular structures. Instructors may also choose to contribute their own DC problems to be added to the system for use by others.

Students may access course shells they are associated with and complete DCs assigned to them by combining textual descriptions of their solutions that they can also augment using a canvas that supports informal box-and-line diagrams. The process of reflection is then facilitated by prompting learners to enter answers to the structured reflection framework questions. The resulting DSs are then archived and made available to instructors associated with the appropriate course shell as well as, with student consent, project investigators.

IV. User Study: Experimental Design

In order to gather formative evaluation feedback from our target learner population, we designed and deployed a focus group user study in spring of 2014. The experimental design was primarily focused on broadly assessing learner perceptions of our overall deployment methodology, initial design challenges, and structured reflection framework. While the set of study participants was purposefully small given our focus, the qualitative dataset gathered from this study (presented in Section V) provides the foundation for drawing valuable insights on our specific approach and reflective design learning, which are discussed in Section VI.

Study participants included six students randomly selected from a set of 16 volunteers drawn from Northern Arizona University's bachelor of science in computer science program. Five students were male and one was female, while all students were in their third year of study in the program. Our study protocol underwent institutional review board approval and all participants were briefed on the study design and signed informed consent forms—there was no reward for participation and no penalty for non-participation.

The study was prefaced by a short introduction, outlining the general steps that participants were expected to follow: First, each participant was provided a DC with instructions asking them to outline a solution to the design problem presented, a phase which lasted 15 minutes. This was followed by participants being provided with an expanded version of the reflection framework appearing in Table I and engaging in structured reflection—this activity also lasted 15 minutes. We concluded with 10 minutes devoted to participants making any additions to or refactoring their initial solutions.

After collecting participant solutions and reflective narratives, the project investigators left the room and participants engaged in a focus group discussion lasting close to an hour. A well-qualified external evaluator unaffiliated with our university moderated this discussion. Our discussion on qualitative results draws on a de-identified transcript of participant feedback provided by the project evaluator as well as the evaluator's report. Details on the DC used in this user study and the questions that prompted the participant discussion appear in the following sections.

A. Design Challenge and Solution

Instructions associated with the DC included prompts for participants to consider possible design concerns, such as how the solution should be functionally decomposed, how functions should interact with each other, the solution's logical flow, and how to store, access, and modify necessary data. Instructions also encouraged participants to use any combination of textual or graphical artifacts, including pseudocode or code snippets, that seemed appropriate in designing a solution.

We drew the DC used in this study from subject matter and learner expertise conventionally associated with a second-semester introduction to computer science and programming course, to ensure that participants were not hindered in their solution by a lack of fundamentals. In addition, we worked toward ensuring that the problem presented a wide variety of design decisions that touched on a number of important areas, such as algorithms, data structures, and functional design, with interesting design dependencies between areas. The problem statement read as follows:

> *You are implementing a text-based version of a battleship game. The board consists of a 10x10 grid. Placed on this board there should be 5 ships in total, each with a number of grid boxes needed to accommodate its size: A patrol boat (size 2), submarine (size 3), a destroyer (size 3), a battleship (size 4) and an aircraft carrier (size 5). These ships should be efficiently placed on the board in a random pattern, where ships can be arranged vertically or horizontally and cannot occupy the same grid space. Each ship must be placed in a continuous sequence of grid spaces according the indicated size. Gameplay should allow for one human player to input a location to "shoot". Clear feedback should be provided to tell the player whether or not the shot was a hit or miss. Additionally, the player should be immediately informed when a ship has been sunk, and also when the entire game has been one.*

In order to provide more robust context for discussing the results of the study, we also present a sample solution to this

DC by a participating learner—this particular solution was solely textual and appears below:

> *The grid would be best implemented as a two dimensional array. The grid would be O's where there is water and a corresponding number where a ship is located (for example, 1 for patrol boat, 2 for sub, 3 for destroyer, etc.). A random function could be used to place the ship by giving a random x,y coordinate and another random binary function to determine if ship is vertical or horizontal. If a collision would happen another random x,y coordinate is chosen. The game would take user input and verify it is accurate and check if that spot contains an O or a number. If it contains a number, replace it with X and say hit otherwise return a miss. If there was a hit the program should run through the array counting all the 1, 2, 3, etc. If a number is not found then that ship has been sunk. To prevent repeat alarms an array containing sunken ships would be needed to verify that the ship sunk last turn and not an earlier turn. The program should then go through the sunken ships and if it is full tell the player that they won.*

We note that this solution does omit some functional details, such as specificity in determining the direction of placement from the randomly determined starting point, and wholly disregards non-functional properties prompted in the problem statement, such as efficiency and clarity of feedback. This foreshadows topics of discussion on lessons learned (appearing in Section VI).

B. Focus Group Prompts

Since our goals with this study included gathering very general feedback from learners, the focus group discussion was encouraged to be wide-ranging and to touch on whatever topics participants felt were important. Nevertheless, our evaluator did provide prompts aimed at ensuring that important topics were discussed within the available time. These prompts were broadly organized and were aimed at gathering feedback on the following topics:

- *Deployment Methodology*: Prompts relating to the overall method of deploying our pedagogical intervention, including the activity sequence and clarity of provided instructions with DCs and the reflection framework;
- *Design Challenge*: Prompts relating to design problems in general as well as the specific DC problem used, including appropriateness for that intended subject matter and possible improvements; and
- *Reflection Framework*: Prompts relating to the questions included in the reflection framework, including quality and perceived value to their learning.

V. User Study: Focus Group Results

The qualitative feedback gathered from the focus group discussion, which lasted approximately 50 minutes, touched on aspects of our entire approach. In some cases, this feedback confirmed our intuition of the challenges involved in design learning and in other cases it provided important insights prompting future refinements and research in this area. This section is focused on presenting participant feedback and draws from our own analysis of the discussion's de-identified transcript as well as our external evaluator's report—a discussion on the insights and lessons we draw from these results appears in Section VI. The following text is organized according to which aspect the material relates to: The overall deployment methodology, issues related to design challenges, and the process of reflection and use of the reflection framework.

A. Deployment Methodology

The initial portion of the focus group discussion centered on the overall methodology of our educational intervention approach and the general sequence of activities involved. Overall, participants recognized the value of explicit reflection in supporting learning and the overall process and activity goals were clear to them. They also identified challenges in separating the activity of solving a DC from that of reflecting over the solution.

Participants felt that the overall process of explicitly engaging in structured reflection over a solution was a unique experience for them that they had not encountered before in the curriculum with one mentioning that "[they have] talked about it a lot but not done something like [this] in any of [their] classes." When prompted to consider the objectives of the activity, they clearly recognized that reflecting over their DC solutions held learning value for them. For example, one participant stated that a main objective was "to see how well we evaluate a system. It's fine to build it, but the questions afterward were like, how much do you really know?" Other participants echoed this sentiment, and added that a main goal was "looking over [their] design choices" and seeing "how [they] could improve everything in part one." In sum, participants recognized the two-fold value of reflection as a way of exploring and learning more about a particular design experience as well as a process that can lead to improved solutions to design problems.

Participants suggested that the process of having them engage in solving the DC and only introducing the reflection framework afterward was not optimal and felt unproductive. They suggested that "part one and part two should be switched", referring to solving the DC and the reflective activity respectively. The reasons they identified were focused along two key concerns: The first is that the questions found in the reflection framework point out important aspects of design and that it might have been helpful to be "passively thinking about the [reflection] questions" while they were solving the design challenge. The second identifies a significant overlap between what they were asked to do for the DC and the structured reflection exercise, particularly in asking them to restate the functional requirements; one student stated that "it becomes repetitive because [they] already did part two in part one" and another felt the overall experience of reflection "would have been more enjoyable if the questions had more meaning [in the context of] the design of the problem." However, these issues did not wholly dominate participant perceptions: for example, one student stated that "[they] really liked the process, even if it was a bit repetitive in general."

Finally, participants overwhelmingly agreed that our time allocations for the user study underestimated the amount of time needed for reflection, with one commenting, for example, that they "would have liked more time on the [reflective process]; it cut off too early." Participants felt the time allocations for other parts of the user study were sufficient.

B. Design Challenge

Another portion of the focus group discussion centered on the general concept of solving a design problem and the specific DC used in the study. Overall, participants verified our expectations of the domain expertise required by the DC we elected to use while pointing out areas for improvement and providing insights into the design process they used and their thinking while solving the problem.

Participants were first prompted to discuss the previously learned knowledge and skills they used in solving the DC. All participants cited their introductory computer science and programming courses as key sources of knowledge that were applied to this problem, while one participant stated that they felt it was "[data structures and algorithms] for [him] because data structures is what really got [him]." Participants also commented on the complexity of the DC problem used and offered suggestions for improvement. They agreed that a "more complex problem" would lead to a more engaging experience and that the problem was too simple for their knowledge level, being more appropriate for "the very beginning" of our program's second-semester introduction to computer science course. Participants also suggested more clarity in the problem statement itself, with one commenting that "[better stating] the directions clearly for what the problem actually is" would improve the challenge, but there were no specific recommendations offered by participants.

The discussion then shifted away from specific domain knowledge, and toward more general considerations: One participant prompted discussion by identifying "problem solving" as a primary skill used in the DC solution process. Another participant stated that his solution selections were "the most obvious answer, but [he thought] that [they were] also the most correct answer." Nevertheless, the discussion did identify that the first answer may not necessarily be the most correct and stated that they utilized a "guess-and-check" approach as their primary method in tackling the problem by thinking about "[implementing] the system and the more [they] thought about it, [they would] start to figure out that maybe [they] could have done this a different way."

C. Reflection Framework

The final portion of the focus group discussion concerned the process of structured reflection and the reflection framework and concluded with a general exploration of the design space participants navigated in their DC solutions. Comments during this portion of the study exposed areas for improvement in our reflection framework while strongly reinforcing our motivating impression that learners in conventionally structured curricula lack robust design expertise. Importantly, the focus group activity ended with a very interesting discussion that had participants share their varying data structure design decisions, which clearly demonstrates the value of group-centric reflection.

There was an overall feeling among participants that the reflection framework could be improved by making the questions more specific to the DC being solved. One participant, for example, stated that "maybe [asking] 'was a matrix the best possible choice?'...because [they were] stuck in the mindset of [it was] only just a matrix: [they were] going to stick with that." The same participant followed up by saying "but if you open up that question, it's like, you know, there could be possibly another way of doing it." Other participants echoed this sentiment, and added that it would be interesting to have the reflection framework guide them along a set of predetermined design options—one said "it's always more fun to figure out different ways you didn't know about before, especially if there is clever tricks and stuff are easier and more efficient to use." Another participant suggested that the solution to this generality challenge they were discussing might be easily addressed by rewording reflection questions; they offered an example of "changing 'what could you do differently?' to 'what are three things you could do different."

Participants expressed an overall feeling that they were somewhat unprepared for the reflective aspect of the user study and some of the specific questions in the reflection framework. At least two of them were not sure what a design notation was. One student was not sure if notation "meant drawing or anything", while the other knew that it "had relation to the software engineering class [they were] in." Similarly, the concept of design principles appeared "vague" to participants, even though they acknowledged that "[they have] gone over it" in one of their courses, but "[they] did not remember what they [were]." Another participant confessed that "coming in, [they] did not know what the difference was between non-functional and functional requirements," which lead to confusion during the structured reflection activity. In terms of non-functional requirements, participants also expressed that they had difficulties in teasing out the non-functional properties that were important in the DC solution; one stated that they were "poised for there to be a lot more [non-functional properties], especially with it being a question and having so many bullet points," but they "didn't really see any when [they] were reading through the problem."

Participants agreed that it would have been helpful to include examples in the framework; one stated that examples would have allowed them to "have a better gauge of what [they are] supposed to be focusing on" and others agreed that "having an example would be good." The use of examples was thought to be particularly important to learners in early stages of their course of study, since they have not yet mastered the terminology used in the reflection framework. A participant stated this argument, saying that “especially before [software engineering] no one is going to know what a non-functional requirement is” and that lower-level students “don’t think about design, [they] just read the problem and start coding” until “in software engineering [they] learn about design.”

The focus group discussion concluded by shifting toward exploring the design space captured in the DC chosen for the user study. The beginning of this discussion entailed participants expressing their feeling that the reflective activity was not particularly effective in helping them consider alternatives; one stated that “[they would] probably change stuff as [they were] implementing,” and that if they found “something easier, of course [they would] implement that one.” Another participant tempered this perspective by stating that they “didn’t think that [they] really needed to change anything,” but that the activity made them “think [they] could change this part if [they] needed to.” Participants also mentioned that they didn’t actively consider certain design aspects during their solution, which became evident during the reflection process. Specifically, “[they] didn’t go through the list [of design principles], they just [came] naturally when solving the problem” and “[they were] not thinking about non-functional requirements during the design because those [were not] important until implementation.”

These comments then prompted an interesting collaborative exploration of the DC’s design space, In terms of data structures, for example, one student stated that they thought about beginning “with a matrix that didn’t hold any values and then implemented several lists, and then like [they] could probably get rid of those lists and implement just the matrix.” There were, of course, variations among participant solutions, such as one solution which “used a matrix of different values, to say whether or not a ship was there—that there was a hit or if it was empty” and another that adopted a “layered approach” with “separate lists [holding] different values” and “two steps of communication.” Another participant suggested that they did not use a matrix, but “an array that held all the ships” which could then be used to “check their health values” as well as “subtract from the health value of a ship” if a ship was hit on the board.

VI. User Study: Insights and Lessons Learned

By reviewing the results discussed in the preceding section, we are able to draw important general insights on reflective design learning and identify promising improvements to our specific approach and associated artifacts. Some of the insights drawn from participant feedback validate our understanding and approach for better integrating reflective design learning into existing curricular structures. Learners, for example, exhibit a strong recognition of the value of reflection in helping them improve solutions and better understand design alternatives. Furthermore, the challenges that participants pointed out they experienced in using our artifacts emphasize the importance of improving how well existing curricular structures support building design expertise.

Most importantly, we draw lessons that are widely applicable in reflective design learning: Participant reactions clearly identify the importance of ensuring a high degree of congruence between learner perceptions regarding the difficulty of a design problem and their perceptions of their own competence. Our observations during the focus group discussion also point out the centrality of collaborative reflection as a way to better engage learners. Of course, we also draw insights on specific improvements to our approach that will inform our future work, such as providing customized reflection frameworks, exploring a parallelization of the problem solving and reflection activities, and investigating mixed-mode deployments of our pedagogical intervention.

The following sections offer more detail on these insights, and follow the same organizational structure as Section V: We begin with insights drawn from feedback relating to our deployment methodology, continue with those stemming from the discussion on design challenges, and conclude with lessons learned by discussing the reflection process and framework.

A. Deployment Methodology

One of the first lessons we draw from this user study is that learners themselves strongly recognize the value of reflection. While pedagogical literature establishes this, learner motivation relies heavily on their own perceptions of utility. Participants in our user study explicitly referred to how useful the reflective activity was as a way of reviewing the specific choices they made while solving the DC problem and exploring the problem’s design space in more depth. Furthermore, they viewed reflection as an important element in improving initial solutions, as a result of having their reflection expose design alternatives they did not initially consider. This recognition that reflection is a valuable activity on the part of our study participants validates the overall focus of our work in using reflection to provide an active and engaging means through which to improve design learning.

Participants did point out that they perceived a significant overlap in the tasks they were asked to do for the DC solution and the tasks involved in the structured reflection activity, which led to an overall sense of repetitiveness. Our intent when asking learners to restate elements of their solution as part of the reflective activity, particularly when asking them questions about the functional and non-functional design decisions they made, is to reinforce their understanding of the solution they had initially conceptualized. This concern, however, is not as critical as preventing learners from perceiving reflection as “busywork” and therefore becoming less invested in the activity. One immediately useful tactic by which this concern could be addressed would be to reduce the overlap between the DC solution and the reflection activity through modifications to the reflection framework: Omitting questions relating to restating requirements, for example, would be highly impactful in reducing the perception of repetitiveness.

More fundamentally, however, this feedback motivates considering a parallel deployment of DCs alongside the structured reflection activity. Providing the reflection framework to learners for use while they solve the design problem, rather than having the two activities happen sequentially, would reduce the perceived overlap. Furthermore, this may be beneficial to learning as it would allow a more immediate integration of the insights drawn through reflection into the DC solution. While we aim to investigate this rethinking of our deployment methodology, we do identify a possible pitfall: Experienced designers naturally reflect over their design decisions as they make them, based on their expertise and ability to predict the ultimate effects of their design choices. Of course, novice designers lack this expertise. Considering a more parallel learning intervention motivates an investigation of techniques for providing more robust scaffolding to novice learners so they gain the necessary experience to effectively meld reflection into their design process.

Participants also expressed concern that the time allocated to the structured reflection portion of our user study was insufficient, despite the relatively straightforward nature of the DC used. While this concern was only mentioned in passing during the focus group discussion, it holds important implications for our overall approach. It is worth mentioning that the time needed for this activity may decrease as learners gain familiarity with the approach, yet we recognize that the time demanded by the reflection activity is critical in whether adopting instructors can easily integrate our work into existing curricular structures. Accurate and practical time estimates are central in allowing educators to plan the use of their in-class time with learners and make informed decisions as to how much course time to devote to reflective learning.

This presents a challenge that resonates with parallelizing the solving of DCs and structured reflection. As an alternative course-integration strategy, we aim to investigate a mixed-mode deployment, which stands out as a promising avenue to pursue: Rather than having learners engage in solving a DC and reflecting over their solution sequentially, we may have learners engage in solving design problems independently (for example, as homework assignments) and then engaging in the reflection process synchronously, alongside their peers in a classroom setting. This approach would provide key advantages: First, educators would have to devote less course time toward design learning, which would fundamentally ease the adoption of our work. Second, a mixed-mode deployment would make the reflective activity more rewarding, as the asynchronous nature of creating the DC solution would provide more time for a deeper exploration of the problem space associated with a design problem. Finally, this approach would better enable collaborative reflection, which resonates with insights discussed in Section VI-C and would better expose design alternatives and nuances that may not have been broadly considered.

B. Design Challenge

A key issue that emerged through our study is the important impact of learner perceptions regarding the match between the perceived difficulty of a DC and their own perceived expertise. In our focus group discussion, participants confirmed our expectations of the expertise required by our chosen DC, which centered on basic algorithm and data structures commonly associated with a second-semester introduction to computer science course. Our participants, however, consisted of students in their third year of study. Our rationale for choosing this DC was to ensure that our study was not hindered by a lack of appropriate expertise, but we underestimated the impact of the disconnect between the design problem difficulty and perceived learner ability.

Participants felt that the problem was trivial given their perceived level of expertise, which seemed to reduce their level of interest in the activity and may have contributed to the perception that elements of the reflective process were just "busywork." Additionally, this perceived simplicity may have had an impact on the quality of the design solutions produced by learners. We note that some participants initially thought that their first design choice for aspects of the problem was really the only possible option, and that they had a hard time seeing the need to consider alternatives. Since the problem was perceived as very basic, this perception may have had an important impact in encouraging participants to only consider the immediately apparent choice and may have discouraged a more thorough exploration of the design space. This opens up interesting future research directions in how the degree of congruence between the complexity of a design problem and perceived learner competence impacts the quality of design solutions and the thoroughness of *post hoc* reflection.

While we identify a clear need to ensure that DCs are perceived as authentically challenging by learners, we also recognize the impact of complex design problems on the time needed to deploy a DC and its associated reflection by adopting educators—even while using a relatively straightforward DC as we did in our study, the overall process consumed roughly 45 minutes. In addition to previously-discussed alternatives, such as parallelizing the solution and reflection activities and mixed-deployments, another possibility that may address this challenge is to consider the use of large-scale design problems that provide a robust context for problem solving, but only explicitly require learners to provide design decisions and reflections over limited aspects of the problem. More fundamentally, this issue exposes a tension between complexity and practicality and motivates future research in identifying the specific characteristics of design problems that provide the appropriate balance between design space complexity—which provides a rich basis for reflection—and realistic time investments for existing curricular structures.

Finally, we note that participant discussion on the DC portion of the user study focused on the fact that learners perceive the reflective process as very effective in exposing the need to consider alternatives to immediately apparent solutions to problems. This further validates our focus on refining techniques for supporting reflective design learning, while also reinforcing the need for providing learners in early curricular stages with an understanding of design processes that is more sophisticated than what one participant referred to as "guess-and-check."

C. Reflection Framework

One concern that emerged during the focus group discussion on the reflection activity was the orientation of the reflection framework toward general design concerns and the fact that it was not specifically tied to the DC used in the case study. Participant suggestions to alleviate this disconnect included going as far as customizing the reflection framework to specific aspects of the design problem's functional and non-functional requirements—in essence suggesting a problem-specific reflection framework. While we think such a close coupling of design problems with the reflection framework compromises our goal of providing a cohesive perspective on design throughout the entirety of the undergraduate curriculum, it does motivate a consideration of specialized reflection frameworks that are more oriented toward specific content areas.

One approach would be to customize which top-level elements of the general framework are used with DCs drawn from specific content areas. This would provide a better match between the questions being asked during reflection and the context of the subject matter that learners are immersed in at the time of the reflective activity. However, we recognize that this may over-constrain our approach and dilute the value of general reflection—after all, the elements of the reflection framework were selected exactly because they are generally applicable and crosscutting. Consider, for example, reflection questions targeted toward non-functional properties: While an explicit discussion of non-functional properties in the context of software design is conventionally first addressed in software engineering-oriented courses, recognizing the importance of non-functional properties is an important element of the design decisions made in earlier courses as well.

A more graceful tactic might be to maintain all top-level elements of the reflection framework, but provide content-specific questions for each of these elements: For example, modifying questions asked regarding non-functional properties might be oriented toward time complexity in data structure modification operations for DCs that are targeted toward early computer science courses rather than the more general questions shown in Table I. This would still allow the reflection process to include all important high-level aspects of design in the context of a specific course, but would also customize the specific reflection prompts to a familiar context. Another tactic for addressing this issue would be to include specific examples aimed at reminding learners of design terminology, which would provide prompts without detracting from the value of engaging in a wide-ranging reflective exercise.

Finally, we draw an important lesson on student perceptions from our observation of the final stages of the focus group discussion: When prompted to discuss their perceptions of the usefulness of reflection, participants initially stated that they did not see a great deal of value in the activity. This impression seemed to particularly hold when asked if reflection helped them identify alternatives to the design decisions captured in their solution. Interestingly, the tone of the discussion grew more positive toward the overall approach as more participants joined in sharing their specific approach to solving the problem, which exposed a greater diversity among solutions than participants had originally perceived.

Importantly, this discussion points out the centrality of collaborative reflection in design learning. While we had originally conceived deployment modes that allowed for individual reflection, this insight motivates a reorientation of our approach to exclusively use collaborative reflection. Collaborative reflection allows for a richer exploration of the design space, as a group of learners is more likely to collectively consider more design alternatives than any one of them could individually. This also provides an important change in perspective, as learners are cast in the role of knowledge creators that may help their peers, which ultimately supports a higher degree of investment and motivation in the learning process. Additionally, this reinforces our emphasis on preserving the outcomes of the reflection process in the form of DSs as a way of creating a body of knowledge that can enrich individual learner reflections when collaborative reflection is impractical.

VII. Conclusion and Future Work

Typically, design learning is only addressed within few isolated curricular points and limited subject matter contexts. This isolation of design learning contrasts with the ubiquitous nature of design and the fact that design decisions are implicitly made throughout the range of artifacts that compose software systems. Our work is focused on addressing this challenge by developing an educational intervention that is modular and easily adopted, so that design learning can be explicitly infused throughout conventional curricular structures. Furthermore, our work emphasizes reflective learning, which is particularly well suited in the context of design.

Early formative evaluation efforts—focused on a focus group user study aimed at gathering qualitative feedback—provide the foundation for a number of insights and lessons on both our specific approach and design learning in general. These insights inform our plans for future work in this area: In the short term, we will be focusing on improving our learning materials to improve student engagement and interest. Long term research directions include an investigation of the impact of congruence between design problem difficulty and self-assessed learner competence as well as an exploration of how solving design problems and reflecting over design solutions can be effectively parallelized in the learning process. Ultimately, our work will focus on how reflective design learning impacts the quality of design outcomes.

Acknowledgment

The authors are grateful to Northern Arizona University Computer Science students in the Spring 2014 offering of Software Engineering. We also gratefully acknowledge the contributions of Magnolia Consulting's Dr. Carol Haden. This research is supported in part by the National Science Foundation under Grant number DUE-1245427.

REFERENCES

[1] M. Fowler, "Is design dead?" in *Extreme Programming Examined*, G. Succi and M. Marchesi, Eds. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, pp. 3–17.

[2] J. Georgas, "Toward infusing modular and reflective design learning throughout the curriculum," in *Proceedings of the 26th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T 2013)*, San Francisco, CA, USA, May 19-21 2013.

[3] L. Vygotsky, *Mind In Society: The Development of Higher Psychological Processes*. Cambridge, MA, USA: Harvard University Press, 1978.

[4] D. Kolb, R. Boyatzis, and C. Mainemelis, "Experiential learning theory: Previous research and new directions," in *Perspectives on Thinking, Learning, and Cognitive Styles*, R. Sternberg and L. Zhang, Eds. Hillsdale, NJ, USA: Lawrence Erlbaum, 2001, pp. 227–247.

[5] M. Prensky, *Digital Game-Based Learning*. New York, NY, USA: McGraw-Hill, 2001.

[6] J. Bruner, *On Knowing: Essays for the Left Hand*. Cambridge, MA, USA: Harvard University Press, 1979.

[7] D. Schön, *Educating the Reflective Practitioner*. San Francisco, CA, USA: Josey-Bass, 1987.

[8] I. Reymen, "Improving design processes through structured reflection: A domain-independent approach," Ph.D. dissertation, Technische Universiteit Eindhoven, 2001.

[9] C. Reigeluth and C. Rodgers, "The elaboration theory of instruction: Prescriptions for task analysis and design," *NSPI Journal*, no. 19, pp. 16–26, 1980.

[10] H. Simon, "The structure of ill structured problems," *Artificial Intelligence*, vol. 4, no. 3, pp. 181–201, 1973.

[11] D. Marples, "The decisions of engineering design," *IRE Transactions on Engineering Management*, vol. EM-8, no. 2, pp. 55–71, June 1961.

[12] C. Alexander, *The Timeless Way of Building*. New York, NY, USA: Oxford University Press, 1979.

[13] J. Jones, *Design Methods*. New York, NY, USA: John Wiley & Sons, Inc., 1992.

[14] T. Moran and J. Carroll, Eds., *Design Rationale: Concepts, Techniques, and Use*. Psychology Press, 1996.

[15] J. Lee, "Design rationale systems: Understanding the issues," *IEEE Expert: Intelligent Systems and Their Applications*, vol. 12, no. 3, pp. 78–85, 1997.