Understanding design iteration: representations from an empirical study

R. S. Adams Center for Engineering Learning and Teaching, University of Washington, USA

Abstract

Design is a cornerstone of the engineering profession and a prominent feature in how we educate engineers and accredit engineering programs. Design problems are often ambiguous, ill-structured, and may have multiple solutions. As a result, a designer’s understanding of the problem or possible solutions evolves through a process of iteration. Iteration is a symbolic feature in design models that represents a process of revisiting and resolving design conflicts. Although iteration is considered an integral part of design activity and a natural attribute of design competency, there is little research that explicitly operationalizes or represents iterative activity. The purpose of this paper is to provide and discuss theoretically meaningful representations of iteration in engineering design. Representations were generated from empirical data from a comprehensive study of cognitive processes in iterative design activity. The utility of these representations is evidenced in their ability to emphasize empirical findings, highlight qualitative trends and patterns of behavior, and distinguish differences in design success and levels of engineering experience. In addition, these representations may be useful pedagogical tools for engaging design students and design educators in discussions about effective iterative behaviors.
Understanding design iteration: representations from an empirical study

Introduction
Design is a cornerstone of the engineering profession, a prominent feature in how we educate engineers and accredit engineering programs, and one way of describing the competency of our engineering graduates as practitioners (ABET, 1998; NRC, 1995; NSF, 1995; NSPE, 1992). Iteration is a fundamental feature of design activity that signifies a goal-directed process of revisiting aspects of a design task in which the goal is a solution that is internally consistent with an understanding of the problem. Iterations mark an awareness that neither the problem nor the goals are well-defined, and are the result of attempts to reconcile ambiguities and contradictions. In cognitive models of design aspects of this process it is described as problem and solution co-evolution (Adams; 2001; Braha and Maimon, 1997; Hybs and Gero, 1992). As such, the mechanisms underlying iterative cycles can be described as transformational and evolutionary processes that mark a designer’s journey from an under-specified starting point to an elusive target goal (Hybs and Gero, 1992). For each adjustment, the designer must analyze not only the effects of the change but also reevaluate the design task. From our own experiences, we refer to iterations as “another pass”, “moving in a new direction”, “the next version”, “inspiration”, “optimizing” or even “starting over”.

Iteration has been found to constitute effective design practice (Adams, Turns and Atman, in press; Bucciarelli, 1996; Radcliffe and Lee, 1989) and provide mechanisms for supporting design innovation (Dorst and Cross, 2001; Suwa, Gero and Purcell, 2000). For example, Suwa et al (2000) found a significant relationship between invention and unexpected discoveries during design sketching. In a comprehensive empirical study of iteration in engineering student design processes, Adams (2001) found that iteration is a significant component of design activity that occurs frequently throughout the design process; and measures of iterative activity were significant indicators of design success (e.g., “effective behaviors”) and greater engineering experience. Examples of effective iterative behaviors include: 1) more time iterating and more iterations, 2) more time in iterative processes that involved a conceptual shift in understanding (transformative processes), 3) more time in iterations triggered by self-monitoring and examining activities, and in iterations that resulted in revisions coupled across problem and solution elements, 4) more time iterating within and across conceptual design and problem setting activities, and 5) a greater awareness of iterative strategies and processes for monitoring, detecting, and resolving design failures. Observations from study data suggest iterative activity may facilitate learning by allowing the designer to continually revisit and reflect upon each aspect of the design task (Adams, Turns and Atman, in press).

Representations of iteration
Few studies operationalize or denote iterative behavior in engineering design, in particular how iteration relates to experience or performance. Representations from studies of design activity indicate iteration as cyclical processes of revisiting previous design decisions and these processes occur predominantly during conceptual design. In a substantial study of engineering student design processes Atman et al (1999) found that seniors made more transitions between steps of the design process than freshmen, and that transition behavior related positively to final solution quality. Representations of these design processes suggest iteration may be described as transitioning backwards to previous design steps. Tjandra (1998) utilized representations for analyzing iteration in design teams and observed both probabilistic or unplanned iterations and parallel task activities of analysis and synthesis; however, no correlation between the quality of the solution and the number of iterations was found. Goldschmidt (1996) created a graphical means to measure design
productivity as the “interlinkability” between conceptual aspects of design. Although greater productivity was not an automatic consequence of a higher ratio of interlinkability, Goldschmidt notes that the technique might be useful for indicating repetitive clarification and evaluation processes. Badke-Schaub and Frankenberger (1999) utilized a graphical framework based on critical situations to study factors that influence collaborative design work in practice. Critical events were defined as situations in which the design process takes on a new direction on a conceptual or embodiment design level. The authors found that critical events accounted for 88% of the situations observed and identified mechanisms responsible for positive and negative outcomes of different critical events.

A question remains: what does iteration look like? Representations of iterative activity may help answer this question. Researchers have utilized graphical representations of data as a mode of inquiry (Chimka and Atman, 1998; Larkin and Simon, 1987) and suggest that representations may increase the variety of questions about educational situations (Eisner, 1997). Representations have also been used as pedagogical devices (Turns and Atman, 2000). This paper was motivated by representations derived from a comprehensive empirical study of iteration (Adams, 2001). In this paper, representations generated from study measures are provided to emphasize and explore iteration in engineering design (e.g., where iterations occur, relative frequency and duration, and patterns of behavior). Representations include timelines of iterative cycles and processes and web diagrams of iterative transition sequences. The utility of these representations is demonstrated in their ability to illustrate theoretically meaningful measures and patterns of iterative activity. The utility of these representations may be extended as educational tools: to educate students about iteration in design and to engage design educators in discussions about improving the teaching of design.

**Extending an empirical study of iteration**

The representations discussed in this paper were generated during a comprehensive study (N=32) of iteration in engineering design (Adams, 2001). The purpose of this study was to 1) empirically explore and identify iterative behaviors in engineering students’ design processes based on a cognitive model of iteration, and 2) compare measures of iterative activity across differences in performance and engineering coursework. The research design was a strategic comparison of freshmen and senior engineering undergraduates and included exploratory and confirmatory components. Pre-engineering freshmen completed the research task prior to enrolling in an introductory engineering course, and seniors completed the task during their final semester before receiving a baccalaureate degree in engineering. The purpose of the exploratory component was to develop and utilize a coding scheme for analyzing iterative activity; hypotheses generated were tested in the confirmatory analysis.

**Methods**

This study utilized a subset from an existing dataset of 50 engineering students solving a complex design problem (Atman et al., 1999; Bursic and Atman, 1997). Eight subjects were selected for the exploratory analysis (4 freshmen, 4 seniors) and 24 subjects for the confirmatory analysis (12 freshmen, 12 seniors). The research method was verbal protocol analysis in which subjects think aloud as they perform an experimental task (Ericsson and Simon, 1993). The experimental task was administered in a laboratory setting. Subjects were given three hours to design a fictitious playground and all requests for additional information were catalogued. Existing data utilized in the iteration study included: 1) protocols previously coded for design step activities, 2) final quality scores based on criteria from expert playground designers, 3) information requested, and 4) background information.
Operationalizing iteration
The framework for coding iterative activity was based on a cognitive model describing underlying mechanisms of iteration as well as schemes for classifying iterative cycles and processes (Adams, 2001; Adams and Atman, 1999; 2000). Attributes of this framework were drawn from a synthesis of research in design and complex problem solving. As illustrated in Figure 1, iteration is operationalized as a goal-directed cognitive process that is triggered by an information processing activity and concludes with a change to a design state (i.e., process, problem, or solution element).

![Figure 1: A cognitive model of iteration in engineering design.](image)

Information processing activities describe how information is being accessed, utilized, and generated. Example triggering activities include monitoring self-understanding or progress, clarifying the nature of the design problem, conceptualizing design elements, and evaluating solution quality. Changes to a design state (the outcomes of an iteration) include redefining problem requirements and evaluation criteria, proposing or modifying new solution elements, and coupled changes across problem and solution elements. Information processing activities that culminate in changes are defined as resolved or successful iterations; situations in which the process does not yield an outcome are defined as unresolved or unsuccessful iterations.

Classifications for successful iterations were coded in terms of **iterative cycles** and **iterative processes**. Iterative cycles are signified by the main outcome of the iteration and codes include: problem scoping, solution revision, coupled cycles in which problem and solution elements are simultaneously revised, and self-monitored cycles in which the iteration is triggered by an explicit plan to revisit a previous design decision. As shown in Figure 1, iterations that connect information processing and decision activities are defined as either diagnostic or transformative processes. Diagnostic processes are defined as incremental revisions in which no major conceptual shift in understanding occurs (e.g., result in corrective actions). Transformative processes are defined as conceptual innovations in which new information is integrated into the process (e.g., result in synthesis or generation actions). For example, iterations that included redefining the problem or coupling revisions to problem and solution elements were coded as transformative; iterations that included only reviewing the problem (without revision) or modifying a solution element (without revising an understanding of the problem) were coded as diagnostic. A complete description of the coding process is provided in previous work (Adams, 2001; Adams and Atman, 1999; 2000). Interrater reliabilities for coding protocols averaged greater than 85% and all differences were arbitrated to consensus.
Because protocols were previously coded for design step activities there was a unique opportunity to combine descriptive and cognitive models of design into an integrated framework for analyzing iterations as *movements located within a design process*. Design step activity codes for the descriptive model are shown in Figure 2 (see Atman et al., 1999; Atman and Bursic, 1998). The links in the idealized web diagram represent iterations as transitions to previous design step activities (e.g., Feasibility to Modeling, Modeling to Gather Information). Iterations can also occur within design steps (e.g., Modeling). This combined framework provides a mechanism for analyzing iterative activity in terms of where iterations are likely to be triggered as well as the direction of an iterative sequence: links begin where an iteration is triggered and the direction of the arrows signifies the goal of an iterative transition sequence.

**Representations of iteration**

Graphical representations of iteration in engineering design were generated in the process of analyzing empirical measures. These representations include *timelines* of iterative cycles and processes and *web diagrams* of iterative transition sequences within a model of design processes. Timelines of iterative cycles and processes display coded behaviors from a chronological perspective and were used to explore the history of iterative activity as well as the relative
frequency and duration of iteration. Web diagrams of iterative transition sequences illustrate relationships between coded design step activities and coded cognitive activities that trigger and resolve an iteration (see Figure 2). These diagrams provide insight into where iterations occur within a model of design and the direction (or goal) of an iteration.

The following sections provide examples of iteration timelines and web diagrams for three subjects. Subject A (a senior) is an example of subjects that received high quality scores and had greater engineering experience. The representations for Subject A illustrate iterative behaviors that correlated with higher quality scores (effective behaviors); and illustrate, although to a lesser extent, patterns for freshmen that received the highest quality scores. Subject B (a canonical freshman) is an example of subjects that received lower quality scores and had less engineering experience. In general, representations for Subject B illustrate a reduction in time spent in effective iterative behaviors as compared to those for Subject A. The representations for Subject B also exemplify patterns for seniors that received the lowest quality scores. Subject C (a freshman) is an example of subjects that received the lowest quality scores as well as had less engineering experience. The representations for Subject C generally illustrate a dramatic reduction in effective iterative behaviors as well as an increase in iterative behaviors that significantly correlated with lower quality scores (ineffective behaviors). Freshmen and senior subjects in this study did not differ significantly across academic and personal backgrounds on the following measures: high school grade point averages, math and verbal scores on standard achievement tests (SAT), and parents’ technical background. Therefore, it is not likely that differences in the representations can be attributed to personal characteristics.

Illustrating iteration: timelines of iterative cycles and processes
Timelines of iterative cycles for the three example subjects are provided in Figure 3. Codes for iterative cycles are listed on the left side of the timelines and the tickmarks represent time engaged in coded activities at that point in time. In the timelines, time is presented as hr:min:sec:msec. Codes for iterative cycles include: Problem Scoping (PS), Monitored Problem Scoping, Solution Revision (SR), Monitored Solution Revision, Coupled Problem and Solution Revision (Coupled), and Monitored Coupled Problem and Solution Revision. Coupled cycles refer to iterations in which revisions to problem and solution elements are occurring simultaneously. Characteristics of coupled iterations observed in the protocols include gathering information on a “just in time” basis, qualifying or quantifying problem requirements by justifying or describing how a solution functions or behaves, and evaluating solutions while clarifying evaluation commitments from multiple perspectives.
A: *High Quality Senior—Total Time Iterating (39.9%), Quality Score (.585)*

B: *Canonical Freshman—Total Time Iterating (29.8%), Quality Score (.409)*

C: *Low Quality Freshman—Total Time Iterating (23.0%), Quality Score (.373)*

Figure 3: Representations of iteration timelines for (a) a senior with a high quality score, (b) a canonical freshman, and (c) a freshman with a low quality score.

The timelines in Figure 3 reveal that iteration occurs frequently throughout the design process (an average of 8 iterations every 5 minutes) rather than at specific points in the process such as optimizing a design solution at the end of the process. The representations also communicate that iteration occurs a significant portion of the time regardless of differences in quality or experience. Freshmen and seniors, respectively, spent an average of 31.4% and 39.8% of their total design time iterating. Comparing across Subjects A, B, and C these iterative cycle timelines emphasize a general reduction in known effective iterative behaviors as levels of design success and engineering experience decrease. These include a reduction in 1) the frequency (and number) of iteration, 2) the levels of coupled and self-monitored coupled cycles, and 3) the likelihood of any self-monitored iterative cycle.

The timelines in Figure 3 also highlight patterns of iterative activity associated with greater success and engineering experience. Comparing from Subjects A to C illustrates a general reduction in iterative problem scoping cycles early in the process. In addition, these cycles appear to be replaced with iterative coupled cycles relatively early in the process suggesting that many students...
(particularly freshmen) did not create a stable representation of the problem prior to developing solutions. The timelines also indicate a relationship between the amount of iterative problem scoping and solution revision cycles. In particular, those who spent a greater portion of time in problem scoping cycles as compared to solution revision cycles received higher scores and generally had more engineering experience. In addition, a pattern is evident in the timeline for Subject A (but not for B and C) in which large “packets” of iterative coupled and self-monitored cycles are closely grouped together. These may be design strategies in which iteration plays a fundamental role.

In addition, the timelines draw attention to a noticeable pattern of iteration at the end of the design process. From the protocols these were observed to be efforts to verify and optimize the quality of final solutions (e.g., verification cycles). For Subject A these verification strategies were more likely to be self-monitored solution revision cycles. Self-monitored cycles are driven by an explicit plan to revisit design decisions and were observed in the protocols to be markers of metacognitive strategies. For Subjects B and C these cycles were more likely to be coupled iterative cycles in which new information was generated and integrated into the task during the final stages of the process. Observations from the protocols suggest these may be efforts to rationalize design solutions by justifying a new understanding of the design task. Finally, by comparing the size of the tickmarks in the timelines it is evident that Subjects B and C were more likely to have iterations of longer duration, whereas Subject A was more likely to have iterations of relatively short duration (average of .68 minutes). As such, this suggests that levels of experience may play an important role in how quickly designers can respond to critical situations.

Timelines of iterative processes for the three example subjects are provided in Figure 4. Codes for iterative processes include Diagnostic and Transformative and are listed on the left side of the timelines. Iterative processes were coded as transformative when revisions involved a conceptual shift in understanding; otherwise, iterative processes were coded as diagnostic. From the empirical study, time spent in transformative iterative processes positively related to higher quality scores and correlated significantly with a higher number of information requests across more categories. Transformative processes also highly correlated with the level of coupled iterative cycles.

The timelines in Figure 4 reveal that the bulk of iterative activity involves transformational processes. This suggests that much of iteration can be characterized as generating and synthesizing information into the design task rather than optimizing relatively stable solutions. The timelines also suggest patterns regarding time spent in diagnostic and transformative iterative processes. Comparing across Subjects, the ratio of time spent in transformative in relation to diagnostic iterative processes approaches unity as the level of success and experience decrease. For Subject A the ratio of time spent in transformative processes is noticeably greater than time spent in diagnostic processes; for Subjects B and C the ratio approaches unity.

The timelines in Figure 4 also highlight differences regarding when diagnostic and transformative processes occur. For the high quality example (Subject A), the timeline shows a high level of transformative processes that decreases dramatically about an hour into the task and a related increase in diagnostic processes for the remainder of the task. Such a pattern seems logical: as an understanding of the problem stabilizes it would be more likely that later revisions would be at a syntactic (e.g., diagnostic) level rather than a semantic (e.g., transformational) level. In other words, for these revisions it would be less likely to require or elicit a conceptual shift in understanding. In comparison, subjects with lower scores and less engineering experience were more likely to spend time in transformative iterative processes later in the design task. From the protocols, large quantities of diagnostic iterative processes early in the process were associated with reviewing the design task and difficulties with bringing new information into the task to guide
design activities. Finally, the timelines in Figures 3 and 4 highlight a difference across final quality scores and experience in the nature of final verification cycles. Whereas Subject A was more likely to spend time in iterative diagnostic cycles at the end of the task, Subjects B and C were more likely to spend time in transformative iterative cycles.

**A: High Quality Senior—Total Time Iterating (39.9%), Quality Score (.585)**

**B: Canonical Freshman—Total Time Iterating (29.8%), Quality Score (.409)**

**C: Low Quality Freshman—Total Time Iterating (23.0%), Quality Score (.373)**

Figure 4: Representations of iterative process timelines for (a) a senior with a high quality score, (b) a canonical freshman, and (c) a freshman with a low quality score.

Overall, the timelines of iterative cycles and iterative processes bring to light empirical findings and reveal patterns of iterative behavior associated with levels of design success and engineering experience. For example, they are useful for emphasizing known effective iterative behaviors, the relative amount of different kinds of iteration in design, and identifying strategies such as final verification loops and early problem scoping activities. As such, these representations highlight the importance of iteration in design as well as effective iterative behaviors that may be useful in the teaching of design.

**Illustrating iteration: Web diagrams of iterative sequences**

Web diagrams of iterative transition sequences within a model of design processes for the three example subjects are provided in Figure 5. The web diagrams illustrate time spent in iterative activities in relation to design activities (e.g., iterating within Modeling and iterating across Feasibility to Gather Information). The percentages shown in the diagrams refer to the amount of total iteration time engaged in that activity. For the case of iterating within a design step, percentages are located within the associated design step symbol. For iterating across design steps, percentages are located on the arrow and the direction of the arrow is towards the goal of the iterative transition sequence.
Figure 5: Representations of iteration web diagrams for (a) a senior with a high quality score, (b) a canonical freshman, and (c) a freshman with a low quality score. Percentages represent percent of total iteration time engaged in that activity. Percentages signified with **(*)** represent known effective iterative activities and those with *(*)* represent known weakly effective activities.

The web diagrams emphasize the variety of possible iterative transition sequences and reveal the significant and positive relationship between the number of iterative transition sequences (and time spent in effective behaviors) and greater design success and engineering experience. For example, the web diagram for Subject A shows 14 different sequences; for Subject B there are 11, for Subject C, only 3. Although the empirical findings identify that a greater number of iterative sequences relates to design success and greater experience, the measure is not a powerful indicator on its own—but rather is limited by the number of sequences present in the web diagrams known to be...
effective (e.g., the amount of time in effective iterative activities). As shown in Figure 5, effective iterative behaviors from the empirical study (signified with "**") include iterations within problem scoping activities, within conceptual design activities, across conceptual design and problem scoping activities, and across implementation and conceptual design activities. Subject A spent 35% of their total iteration time in effective activities; whereas Subjects B and C spent 20.4% and 29.6% of their total iteration time. Also, Subject A spent time in 7 of a possible 9 effective iterative activities, whereas Subjects B and C spent time in 4 and 3.

The web diagrams also highlight the trade-off between time spent in known effective iterative activities and iterative activities positively associated with success but not statistically significant (weakly effective activities). Weakly effective iterative activities are signified with a "*" in Figure 5 and examples include time spent iterating within Modeling and Feasibility, and from Modeling to Generate Ideas. Subject A spent approximately equal times in effective and weakly effective activities (35% and 37.6% respectively). Subject B spent more time in weakly effective as compared to effective activities (37.2% and 20.4%), and Subject C spent almost twice as much time in weakly effective as compared to effective activities (58% and 29.6%). These trends suggest that the process of acquiring design expertise may be associated with replacing weakly effective strategies with considerably more effective strategies.

A comparison across web diagrams indicates a general increase in known ineffective behaviors as quality scores and engineering experience decrease (e.g., iterating from Generate Ideas to Problem Definition). Similarly, the web diagrams clearly reveal a relationship between an excessive level of iterating within Modeling and lower quality scores and less engineering experience. Although time spent iterating within Modeling was found to be a weakly effective iterative activity, an excessive level was associated with lower quality scores. For example, Subject A spent 13.2% of their total iteration time within the Modeling design step whereas Subject C spent 58% of their total iteration time.

Finally, the web diagrams highlight the relationship between where iterations are triggered and the goal driving the iterative activity: the goal of iterative sequences is more likely to be related to problem scoping activities, in particular transitioning back to Gather Information. This indicates that problem scoping activities represent not only a significant design goal but also occur throughout the design task in qualitatively different ways as solutions are developed. An interesting finding suggested in the empirical study but best represented in these web diagrams is a pattern of iteration that can be characterized as a conversation across representational spaces: between conceptual design and problem scoping, communication and conceptual design, and communication and problem scoping. Aspects of this iterative activity may be conceptualized as design discourse (Adams, Turns and Atman, in press; Mandershetty, 1995). For example, Mandershetty (1995) created a cognitive model of design in which problem and solution representations developed during conceptual design activities set up a universe of discourse that encourages the generation of novel ideas or design breakthroughs. Observations of such conversations in the protocols were described as problem scoping in context. As an example, a student begins with an abstract sense of the design constraint “be safe” and as they move through the design process and develop solutions they generate an understanding of safety in specific solution contexts and revise solutions based on this new understanding. In the process they elaborate or expand a conception of safety at a more generalizable level which can then be used to guide the improvement of other solution elements for which safety might be an important constraint.

Such a dialectic is indicative of more expert like strategies found in other complex problem solving domains and is believed to be a hallmark of expert task performance. In the context of expertise in reading and writing, Scardamalia and Bereiter (1991) developed a model of skill acquisition as a
dialectic process between particular and general conceptualizations. In a study of writing as a complex problem solving process Bryson et al., (1991) found that experts interpret the significance of the topic on a more abstract level and transform it so that it can be placed in a more meaningful epistemological perspective. The authors describe this as a dialectical interaction between content and rhetorical goals in which a representation of the problem evolves recursively as cognitive operations bridge the gap between initial and final states. In the context of expert actors, Noice and Noice (1997, pg. 69) remark “throughout, it was obvious that the participant (the expert) examined the written text for the purpose of turning it into a living conversation.”

Conclusions

Representations of iterative activity are effective and useful mechanisms for communicating theoretically meaningful empirical findings and revealing qualitative characteristics of iteration in engineering design related to performance and engineering experience. These representations clearly indicate the extent to which design is an iterative process as well as the variety of iterative strategies designers utilize. Similarly, activities captured in the representations help articulate the meaning of empirical findings from a confirmatory study of iterative processes in design. From a theoretical perspective, qualitative patterns evident in the representations illustrate design iteration as a conversation across representational spaces. As a hallmark of expertise in the solving of complex problems, aspects of dialectic iterative activity may be useful as markers of design learning. The means for capturing these dialectic patterns may be extended to support similar studies in other complex problem solving domains.

From a practical perspective these representations have high utility for encouraging a dialogue on iteration in engineering design. For example, design educators could use the representations presented in this paper to engage their students in a conversation about the role of iteration in design and effective iterative activities. The representations also suggest that iterative activity should be strongly encouraged in the teaching of design. Educators could use these representations to justify pedagogical decisions such as increasing opportunities for students to iterate frequently in their design activities, as well as offering instruction in iterative strategies and promoting an awareness of iteration as a successful design strategy.

Acknowledgements

This research was made possible in part by a National Science Foundation grant RED-9358516, the Engineering Coalition of Schools for Excellence in Leadership and Education (ECSEL), a National Science Foundation Engineering Education Coalitions program, as well as a grant from the GE Fund. I would also like to thank all of the students who participated in this study, Cindy Atman who allowed me to use her data to tackle a unique and substantial project, Jennifer Turns for her irreplaceable insight, and Jana Littleton for assisting in the coding of the transcripts and improving the coding scheme.
References


