Instructional Design is to Teaching as Software Engineering is to Programming

Austin Cory Bart and Clifford A. Shaffer
Virginia Tech
{acbart, shaffer}@vt.edu

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer Science Education

Keywords

Instructional Design; Teaching; Software Engineering; Dick & Carey; Formal Methods

1. SUMMARY

This special session will explore practical results from the educational theory of Instructional Design (ID), with particular focus on the widespread similarities between a process for creating successful courses and a process for creating successful software. We present a small set of specific practices that should be easy for CS educators to adopt. In particular, the session will cover the popular Dick & Carey model, meant for beginners to ID. This model helps instructors rigorously define *who* they will teach to, *what* they will teach, *how* they will assess, and (only then) *how* they will teach. The approach is parallel to Software Engineering techniques such as Test-Driven Development, Requirements Engineering, and Iterative Development.

The session will be a blend of presentation, participation, and assessment. Participants will work in small groups both to foster discussion and to provide learning support. The content of the presentation will particularly focus on how the model can be applied practically. It is our hope that attendees, whether new to teaching or experienced, will adopt or be influenced by the model in order to approach their courses with the same rigor they apply to software development.

2. OBJECTIVE

Instructional Design is the iterative, systematic design of effective learning experiences by following a coherent process that gives measurable results. Its history stretches back into the 1930s [2], but there has been surprisingly few published cases of its use in Computer Science Education. Going the other way, there is a body of research on how Software Engineering techniques can be applied to ID [3, 4, 6]. Within CS, what little research that exists focuses on pedagogical tools and tactics [1, 5] rather than holistic, cohesive processes.

Figure 1 gives an overview of the phases of the Dick & Carey Model. Notice that the actual development and testing of instructional materials and strategy is saved for the final phases of the model, rather than the beginning. Instead, instructors begin by analyzing their "users" (the learners) and their instructional goals, then determine concrete objectives and measurements (tests), and finally focus on design and development. Similar to how Software Engineering does not dictate architecture but rather focuses on process, such as how to do documentation and testing, ID promotes process for things like documentation and organization over constricting your pedagogy. ID is compatible with other popular educational theories, including Constructivist Learning, Active Learning, and Collaborative Learning.

We will present a number of beneficial strategies with specific implementation recommendations, including:

Course Alignment (≈ Quality Management) ID leads to explicit definitions for learning objectives, assessments, and course activities, making it clear how components fit together.

Templated Instruction (≈ Design Patterns) Development of instructional strategy and material can be guided by templates.

Assessment Evaluation (≈ Unit Testing) Instructional faults can be pinpointed, and further isolated and resolved by using techniques such as Item Response Theory to increase the coverage and validity of the assessment instruments.

Course Documentation (≈ Design Documentation) Good instructional design practice is self documenting. Collaborators can be given more than the course materials. They can get instructional strategy, assessments, and the rationale behind the instruction.

This session will be organized as a blend of tutorial and hands-on, small-group exercise. We will introduce a new, practical educational theory to attendees, many of who may be unfamiliar with it. ID should resonate with instructors that have an engineering background and yearn for similar structure in their design of learning experiences.
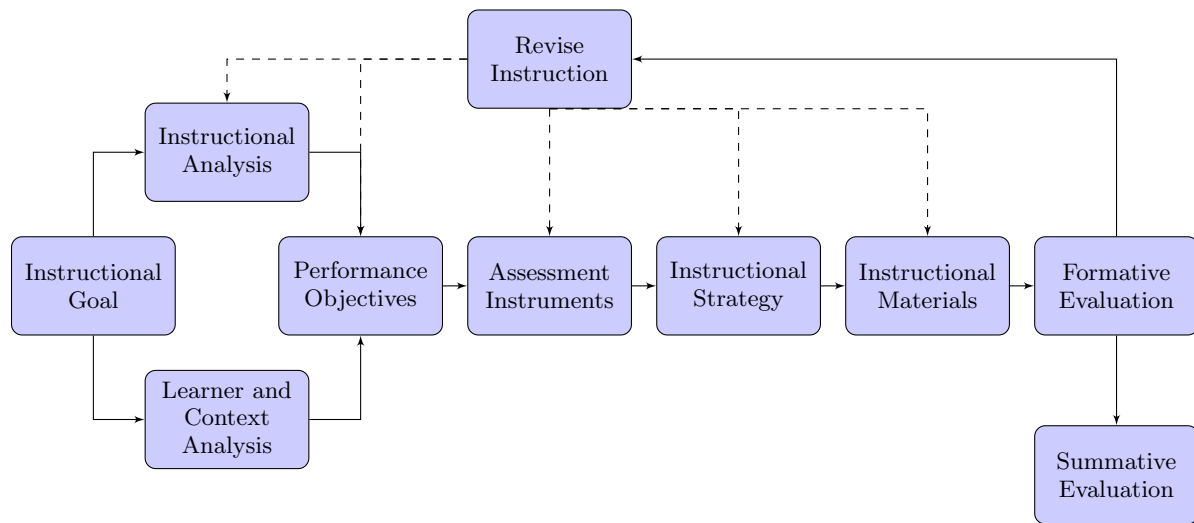
Figure 1: The Dick & Carey Model of Instructional Design

3. OUTLINE

Introduction: (15 minutes) Overview of the ID model, its pros and cons, and practical strategies for using it.

Case Study: (15 minutes) Walk through of the Dick & Carey model and our experiences using it.

Practice Problems: (20 minutes) Small group exercise. Attendees will get to apply the model to solidify their knowledge.

Post-Test: (15 minutes) Attendees will be given a brief, optional "post-test" to help them identify if they learned the material of the session.

Discussion: (10 minutes) Attendees will discuss how the model could be applied to their own classrooms, and their reaction to this special session.

4. EXPECTATIONS

The expected audience is computer science instructors, although anyone interested in educational theory should find the session useful. No prior experience with ID is expected. This session will cover an overview of the Dick & Carey model and demonstrate how it can be used in Computer Science classrooms at any grade level (K-12, undergraduate, etc.), any experience level, and any setting. We expect that instructors with prior knowledge of ID will also benefit from exposure to this concrete method.

The session will cover how instructors can get more information about the model, to promote long-term transfer of the material. Feedback will be solicited from attendees using a survey, and there will also be a pre- and post- assessment (both meant as a formative evaluation of the materials and to guide attendees' learning).

5. SUITABILITY FOR A SPECIAL SESSION

The presenters believe that this session is particularly suitable as a Special Session rather than a paper or panel because it requires hands-on experience and sufficient time to do so. Although a workshop would be a suitable alternative to this format, we suspect that few attendees will be sufficiently knowledgeable of Instructional Design to be willing to invest money in attending it. But we hope that after initial exposure, instructors will be motivated to learn more. The Special Session format allows us to provide the SIGCSE community with a low-stakes introduction to a powerful new technique that could revolutionize the way we design our lessons.

6. ACKNOWLEDGMENT

7. REFERENCES

[1] D. Damian, A. Hadwin, and B. Al-Ani. Instructional design and assessment strategies for teaching global software development: A framework. In *Proceedings of the 28th International Conference on Software Engineering*, ICSE '06, pages 685–690, New York, NY, USA, 2006. ACM.

[2] W. Dick, L. Carey, and J. O. Carey. The systematic design of instruction. 2005.

[3] I. Douglas. Instructional design based on reusable learning objects: applying lessons of object-oriented software engineering to learning systems design. In *Frontiers in Education Conference, 2001. 31st Annual*, volume 3, pages F4E–1–5 vol.3, 2001.

[4] I. Douglas. Issues in software engineering of relevance to instructional design. *TechTrends*, 50(5):28–35, 2006.

[5] S. Hadjerrouit. Learner-centered web-based instruction in software engineering. *Education, IEEE Transactions on*, 48(1):99–104, Feb 2005.

[6] S. Tripp and B. Bichelmeyer. Rapid prototyping: An alternative instructional design strategy. *Educational Technology Research and Development*, 38(1):31–44, 1990.