An Instructional Design Case Study for Computer Science Education

Austin Cory Bart, Eli Tilevich, Clifford A. Shaffer, Dennis Kafura
{acbart, tilevich, shaffer, kafura}@vt.edu
Virginia Tech

ABSTRACT

Abstract will go here

Categories and Subject Descriptors

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

General Terms

Design, Human Factors, Reliability, Experimentation

Keywords

Instructional Design, Robert Gange, Dick & Carey

1. INTRODUCTION

Instructional Design is the systematic design of effective Learning Experiences. Its history stretches back into the 1940s.

However, there has been few reported cases of its usage in Computer Science Education.

This paper describes a specific, popular Instructional Design model through a Computer Science Education case study.

1.1 Roles

An Instructional Designer is expected to be the advocate for the learner. They are, first and foremost, dedicated to providing an optimal learning experience. The instructional designer may or may not be responsible for actually physically teaching the learners (indeed, some instructional materials are designed without any physical instructor at all).

1.2 Models of Instructional Design

A number of competing models ADDIE AGILE

In this case study, we use the Dick & Carey model of Instructional Design, originally published in 1978 but continually updated and refined since then. Figure 1 gives an overview of this model. The model is meant to be iterative.

One of the most influential[**?**] instructional theories is Gangé's Theory of Instruction, a complex Cognitivist model that categorizes different types of learning outcomes (e.g., "Verbal Information" such as reciting the definition of a term, "Attitudinal" such as choosing to not smoke) and breaks down the process of learning into nine coherent steps. Gangé stresses that different learning outcomes require very different instructional implementations of the general instructional model. This instructional theory informs the design of complete Instructional Design models such as the Dick and Carey model [**?**]. Although different ID models suggest different approaches to creating instructional materials, those materials are all typically patterned after Gangé's instructional events.

2. THE CASE STUDY

In this section, we will describe each phase of the D&C model and our results using the model. We took on the role of instructional designers to create new content for an introductory computing class on "Computational Thinking" for non-Computer Science majors. The students in this course are expected to have no prior programming experience and are taking the course as part of the university's general requirements. They come from many different disciplines, including the arts, sciences, engineering, agriculture, and more. Because the curriculum is largely designed from scratch, there are many opportunities for instructional design.

When reviewing our results, keep in mind that the model is meant to be iterative, and any results we have generated are the result of several iterations. Any work presented is therefore neither completed nor prototypical.

2.1 Identify Instructional Goal

The first step in the Dick & Carey Model is to identify the instructional goal(s) that you want to accomplish. Critically, it does not describe anything about how the instructor will teach the material – only what learners are expected to be able to do at the end of instruction. This goal can come from a variety of sources. The most obvious source is a department- or instructor- mandated curriculum, which may explicitly lay out what is expected of students. Goals can also come from a perceived gap in learners existing ability. In the full Dick & Carey model, it is possible that instruction is not needed. For instance, instead of teaching employees how to use complex software, the user interface of the software could be streamlined instead. Because instructional design can be a very time and energy consuming
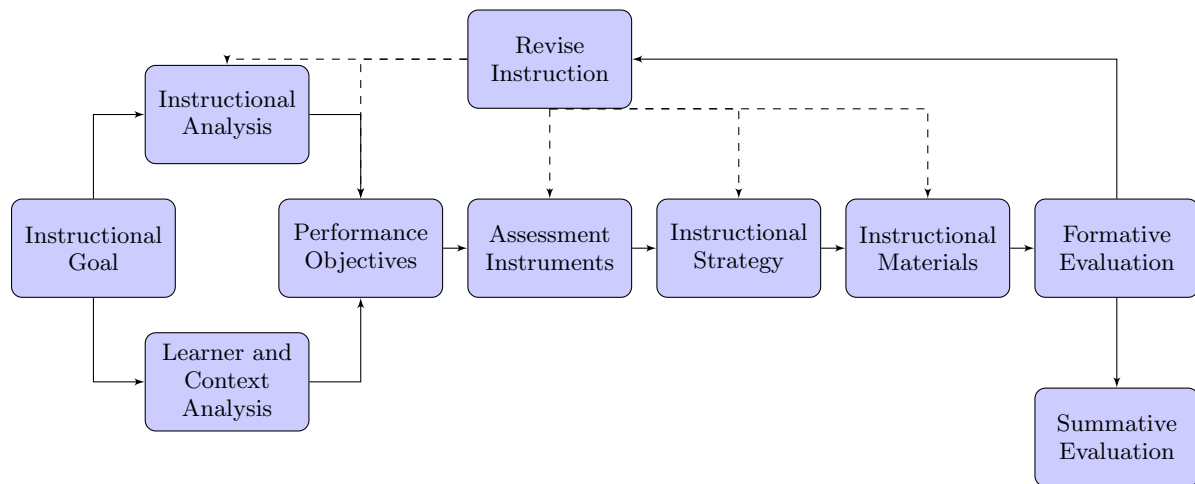
Figure 1: The Dick & Carey Model of Instructional Design

process, it is important to find evidence (either quantitative or qualitative) of the need for instruction.

A complete goal statement should describe: the learners; the performance context; the action that learners should be able to do in the performance context; and the tools that are available to the learners in the performance context. A major, common misstep in writing an instructional goal is to create a *fuzzy goal*, usually describing an abstract, unobservable outcome using verbs such as "appreciate" or "know". To correct a vague goal, the designer should begin by writing down the goal. Then, they list examples of what people do to demonstrate that they have achieved this goal. The designer can use this list to iteratively develop more concrete goals that will lead to observable outcomes.

In this case study, we wanted students to better understand variables and program state. The instructors and teaching assistants of the course repeatedly mentioned this as one of the weakest areas for students, despite being one of the most important conceptually. Many students struggled to write basic programs because of their inadequate understanding of how to use variables. Our final goal statement is:

An Undergraduate, non-CS major (**Learners**)...
... *Given source code* (**Performance Context**) ...
... *Will be able to identify the name, value, origin, and type of a variable at any given step within the code* (**Action**)...
... *Taking advantage of any resources from the internet and their Integrated Debugging Environment* (**Tools**).

2.2 Conduct Instructional Analysis

The next step is conduct an Instructional Analysis, which will spell out exactly what the successful learner will do in the performance context. Like the Instructional Goal, it does not describe how to teach the material, although it is easy for practice-oriented thinkers to accidentally start diagramming the lesson instead. The Instructional Analysis first requires the designer to classify the goal statement based on the the kind of learning it requires. Then, the designer identifies and organizes the *subskills* involved in the skill being learned, and places them into an Instructional Analysis Diagram.

Dick & Carey describe 4 learning outcomes, based on the work by Gagné. **Verbal Information** requires the simplest form of learning: it represents simple, declarative knowledge to be memorized and recalled (e.g., "What is the worst-case runtime of the major sorting algorithms?"). Very little presentation is usually needed for Verbal Information, although students may find it useful to practice such material. **Intellectual Skills** are more complicated problem-solving tasks that demand some level of deeper cognition (e.g., "Write a program to compute the sum of a list of numbers"). Many types of knowledge in Computer Science require Intellectual Skills, but range greatly in complexity. **Psychomotor Skills** are much rarer in Computer Science, describing the coordination of mental and physical activity (e.g., "Type the alphabet using a keyboard"). **Attitudes** is the fourth type, describing goals where the learner makes a choice (e.g., "Choose to persevere in the face of segfaults"). Gagné also identified a fifth domain, **Cognitive Strategies**, but Dick & Carey includes this as a particularly ill-structured form of Intellectual Skills. Readers familiar with Bloom's Taxonomy will find these outcomes similar – mappings exist between these lists, suggesting their rough equivalence. It is up to the designer to choose their preferred schema.

Once the goal is classified, the instructor performs a Subskill Analysis to concretely describe the goal and its components. Intellectual Skills and Psychomotor Skills is a typically straightforward process of creating a flowchart that breaks down each step into multiple levels. Much as you would decompose the steps of a program, you decompose the skills into increasingly simpler steps. Unlike a programming language, however, it is strictly up to the designer to decide when to terminate the iterative decomposition. It is also possible to have branches and cycles, as is typical for flowcharts. Some of the most basic subskills can be considered "Entry Skills" – the learner can be expected to already have mastered them before the instruction begins. These substeps might, in turn, involve Verbal Knowledge, which is denoted with a triangular "V" node. Verbal Information is much simpler to diagram, although the designer can cluster the topics if they want; in the runtime analysis example from before, for example, you might group the algorithms

Figure 2: The Abridged Skill Analysis Diagram
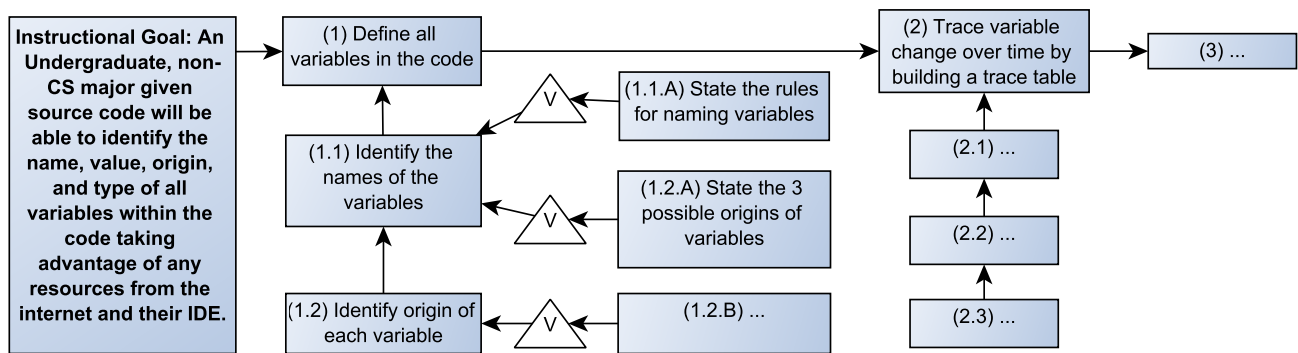
by their Big O values. Attitudinal Skills can be considered a special form of the other skills: to resolve it, simply consider what skill the behavior in the goal is most akin to.

In this case study, our instructional goal is best classified as an Intellectual Skill. Figure 2 is an abridged version of the Instructional Analysis Diagram created for the goal. The complete diagram is available in the online appendix.

2.3 Conduct Learner and Context Analysis

During the next phase, you will need to identify properties about the learners, the learning context, and the performance context. There are many factors involved in learner and context analysis.

Dick & Carey suggest 8 crucial factors: (1) entry skills, (2) prior knowledge of the topic area, (3) attitudes toward content and potential delivery system, (4) academic motivation, (5) educational and ability levels, (6) general learning preferences, (7) attitudes toward the organization giving the instruction, and (8) group characteristics.

2.4 Write Performance Objectives

For every Writing performance objectives is a simple, but repetitive, phase.

2.5 Develop Assessment Instruments

There are four kinds of assessments.

2.6 Develop Instructional Strategy

As software developers, it is extremely tempting to

2.7 Develop and Select Instructional Materials

Instructional Design Models such as Dick & Carey usually delay the design of instructional materials till the end of the entire process, focusing initially on analysis of the learners, the performance context, the instructional goal, and a host of other critical factors. At that point, a content delivery system is chosen based on its educational affordances and suitability for the learning experience being developed. This delay reflects the typical role of an Instructional Designer as a *consumer* of instructional technology – they survey the available techniques and technology (e.g., traditional lecture, essay writing, websites), and choose the one that fits their need.

2.7.1 Gagné's Instructional Events

(1) **Gain Attention:** Stimulate the learner and motivate them to engage with the material.

(2) **Describe Objectives:** Inform the learner will be able to do because of the instruction.

(3) **Stimulating Recall of Prior Knowledge:** Remind students of their prior knowledge.

(4) **Present the Material:** Using text, graphics, etc., give the instructional materials.

(5) **Provide Guidance:** Give instructions on how to use the instructional materials.

(6) **Elicit Performance:** Have the learners actively apply the new knowledge.

(7) **Provide Feedback:** Show the learner what they did well and poorly on.

(8) **Assess Performance:** Evaluate the learner on their performance

(9) **Enhance Retention and Transfer:** Generalize the instruction by showing similar applications.

Figure 3: Gangé's Nine Events of Instruction

Phase	Student	Pretest	Posttest	Difference
1-1	#1	37.8%	100%	+62.2%
1-1	#2	51.8%	100%	+48.2%
1-1	#3	35.7%	84.6%	+49.1%
SG	#1	21.7%	93.3%	+71.6%
SG	#2	54.6%	100%	+45.4%
SG	#3	64.3%	91%	+26.7%
SG	#4	32.7%	80%	+47.3%
SG	#5	69.4%	91%	+21.6%
	Average	45.9%	92.5%	+46.6%
	StdDev	7.5%	16.6%	16.4%

Figure 4: Student Performance during Formative Evaluation

Gangé's nine events are summarized in Figure 3, taken indirectly from[**?**]. This model is described as an ordered, one-way sequence; however, in practice, some steps might be repeated and cycled between. For instance, very negative feedback might prompt the learner to return from step 7 to step 6. Modifications of the model have also been discussed in the literature. For example, to adapt for more Constructivist tastes, one could swap steps 4 and 6, so that students have the opportunity to engage with the materials before receiving instruction.

2.8 Conduct Formative Evaluation

2.9 Revise Instruction

2.10 Conduct Summative Evaluation

3. CONCLUSIONS

4. REFERENCES