



DOLFINITY: A NEW MODEL - FOR BLENDED LEARNING -

Kieran Mathieson

Online courses have many forms. Some are electronic correspondence courses. Others are lecture/discussion courses, with videos, and forums. Some courses use tools like SecondLife to simulate a physical space.

How to choose the best form for a course? One approach is to study research on learning, and use whatever mix of technology and face-to-face instruction is most effective.

This paper examines a model for skill-based courses, like algebra, writing, and computer programming. The model helps skill learning be more effective and efficient, for both students and faculty. It is blended (partly online, and partly face-to-face), and particularly suited to flipped courses (Bergmann and Sams, 2012). The model is called Dolfinity, for reasons explained below.

There is an associated business model as well. Traditionally, textbook authors get a small fraction of textbook revenue, have limited control over content, and can only write for large markets. Students pay high prices.

The Dolfinity business model lets authors create online textbook-like-things (called dolfins), sell them at whatever price they wish, keep most of the revenue, and retain control over content. Authors can offer low prices, and still make a liv-

ing from their work. They can afford to write niche books, even books for one university or school district.

The Dolfinity model is about learning skills. Let's begin by talking about skills and skill learning.

Skills

The goal of a skill course is performance. Students create artifacts, like poems, engineering designs, and lesson plans. Other tasks include repairing an artifact (e. g., debugging a program), and adapting an artifact (e. g., setting a Shakespearean play in modern times).

Competent performers have declarative knowledge, or facts, like how to spell “declarative.” They also have process knowledge, knowing how to use facts to meet performance goals.

Some process knowledge is embedded in patterns. They are solutions that competent performers find useful. For example, rhetorical patterns include metaphor, antithesis, and the rhetorical question. Computer programming patterns include model-view-controller, and the singleton.

Other process knowledge is about problem solving strategies. Some strategies are domain specific, like “to find a program bug, compare the output the program should produce, with the output it does produce.” Other strategies are general aspects of critical thinking, like “identify criteria for judging success.”

Learning skills

Problem solving strategies are relatively difficult to learn. In programming, Robins, Rountree, and Rountree (2003) suggest that:

. . . basic program planning rather than specific language features is the main source of difficulty. (p. 154)

Consequently:

A strong movement in computing and informatics education considers programming to be an application of skills in problem solving. (Pears, Seidman, et al., 2007, p. 207)

Most programming textbooks are organized around language features. There will be a chapter on variables and expressions, another on if statements, another on loops, and so on.

A programming book with a problem solving orientation would be organized around tasks, such as summarizing data, or implementing a map in a role playing game. Language features would be introduced as necessary to complete tasks. This is sometimes called “outcome-driven learning.”

The task-oriented approach lends itself to “deep learning,” a general term for helping students learn processes, as well as declarative facts. A key is to carefully choose the tasks students complete. Students encounter important strategies when the tasks they are given lend themselves to solution using those strategies.

Further, choosing multiple tasks completed with the same strategy helps students generalize across contexts. For example, the solutions to two dissimilar programming tasks might both involve decomposition (breaking a problem into simpler pieces). Decomposition is one of the “Big Ideas” (Wiggins and McTighe, 2005) of programming. That the same strategy can be used for many tasks should be explicitly explained to students, who may not make the inference themselves.

Metacognition helps students examine their strategies. Students reflect on how they worked on a task (Bransford, Brown, and Cocking, 2000). For example, a pair of students might talk about their approaches to solving the same physics problem (Fagen, Crouch, and Mazur, 2002).

Students can’t learn skills without practice. Students should get formative feedback on their work (Bransford, Brown, and Cocking, 2000). This is distinct from summative

feedback, that is, grades. Grades assess how well students learned in the past. Formative feedback helps with current learning, that may be assessed in the future with a summative exam.

Formative feedback is usually a list of specific things a student can do to improve his/her artifacts. Some of the feedback should tie the student's work to process strategies, and to the field's Big Ideas. For example, if a novice programmer does not indent code correctly, s/he should be reminded that indenting helps achieve program readability, another of programming's Big Ideas.

Ideally, students would get a chance to improve their solutions, and submit them again. Take this approach further, and we have mastery learning (Guskey, 1997), where students must demonstrate their understanding of one topic before they can move on to the next.

Dolphins

A dolphin is a combination textbook-workbook-feedback-assessment Web site. It replaces a traditional textbook for a skills course. Dolphin is more-or-less an acronym, for deep, outcome-driven learning, feedback intensive. The n stands for nothing. It just rounds out the acronym.

(If you can think of a better name after you've read the rest of this paper, let me know.)

Dolphins are outcome-driven. Figure 1 shows the table of contents for a dolphin about basic Web design, available at <http://corecogs.com>:

⊕	A Web page with text
⊕	A page that interacts
⊕	A page with images
⊕	A page with tables
⊕	Web pages with links
⊕	Web page layout
⊕	Reuse a design
⊕	A production site
⊕	A site with widgets

Figure 1. Table of contents for a -dolphin on making basic Web sites -

The dolphin helps students learn HTML, CSS, and JavaScript. However, none of the those terms are in the table of contents, because they are technologies, not outcomes. “A Web page with text”—the first chapter—is an outcome. It covers a little HTML and CSS, only that needed to meet the goal.

Figure 2 gives the table of contents for the first chapter. Towards the bottom, you’ll see a lesson on “Writing for the Web.” Figure 3 shows the contents of that lesson.

A Web page with text	
■	The structure of a Web page
■	Patterns
■	Basic HTML tags
■	Simple font tags
■	Adding style with CSS
■	Too much information!
■	Patterns again
■	Making lists
■	Writing for the Web
■	eMe: Some text pages
■	Exercises: A Web page with text

Figure 2. Table of contents - for “A Web page with text” -

1. Where are we?
2. This lesson's goals
3. The Web is text (mostly)
4. Who are you writing for?
5. A passel of guidelines
1. Less text
2. Active voice
3. Less happy talk
4. Eliminate instructions
5. Write for scanning
6. Use lists
7. Sans serif
8. Color contrast
9. Text size
6. Going deeper
7. Summary
8. <u>What now?</u>

Figure 3. Table of contents
for the page
“Writing for the Web”

The lesson is not about technology, but about writing. It covers nontechnical practices needed to meet the goal: Making a text Web page.

DOLFINITY PRINCIPLE 1:

Dolphins are organized around outcomes.

Another chapter is about reusing existing Web designs for a new Web site. Figure 4 shows the chapter’s table of contents. The chapter is entirely about process. No new technology is introduced.

Reuse a design
■ The goal
■ Getting ready
■ Find a design
■ Change layout
■ Change fonts
■ Change graphics
■ Add standard content
■ Build out
■ Renata and CC reuse a design
■ Exercises: Reuse a design

Figure 4. Table of contents for the chapter “Reuse a design”

DOLFINITY PRINCIPLE 2:

Dolphins are fact light, and process heavy.

One of the lessons in Figure 4 is “Renata and CC reuse a design.” Figure 5 shows part of the lesson. Renata and CC are types of pedagogical agents (Lester, Converse, Kahler, Barlow, Stone, and Bhogal, 1997). In a dolphin, they are like characters in a novel. Students watch them work on a Web site, seeing their successes and failures. The most important part of deep learning is the doing, and the feedback. Exercises are central to dolphins; the Web site dolphin has 161 of them. Some are embedded in the content, some are gathered in exercise sets. Most exercises involve artifact creation.

DOLFINITY PRINCIPLE 3:

Dolphins have many exercises.

Students enter their solutions in dolphins, and ask for feedback (Figure 6).

Reviewers evaluate solutions, and give formative feedback comments. Student can correct errors, and submit again. Students earn badges for completed exercises. Figure 7 shows a completed exercise with reviewer feedback.



CC

Makes sense.

There's a heading for the block: Day 1. Make it an `<h1>`!

```

<h1>Day 1</h1>
<p>Blah blah blah</p>
<p class="day_box_link">
  <a href="??">More...</a>
</p>

```



Renata

OK, but this `<h1>` is going to look different from the `<h1>` in the page title.



CC

So let's give this one its own class, so we can make a CSS rule for it:

```

<h1 class="day_box_heading">Day 1</h1>
<p>Blah blah blah</p>
<p class="day_box_link">
  <a href="??">More...</a>
</p>

```

Figure 5. Pedagogical agents

Exercise: Ordered list of dog breeds

Find a list of dog breeds on the Web. Create a Web page with a list of five of your favorites. List them in your order of preference: favorite first, next favorite second, etc. Upload your page to your server, and add a link below.

Your solution

My solution is at <http://renata.com/coredogs/dog-breeds.html>

Edit

Request review?

Yes

Make public?

No

Figure 6. Submitting a solution for an exercise -

DOLFINITY PRINCIPLE 4:

Dolphins users get formative feedback.

Recall that The Web site dolphin has 161 exercises for each student. For a class of 40 students, that is a lot of feedback, even if half of the exercises are optional. However, if we are serious about helping students learn skills, giving formative feedback is not optional. Further, it cannot be given by a computer. Computers cannot assess how well, say, a business letter implements rhetoric’s Big Ideas.

A key to dolfinity is an efficient workflow for giving feedback, implemented as “clickable rubrics.” When an author creates an exercise, s/he also creates a rubric, that is, a set of standards for determining the correctness of a solution. Figure 8 shows an example. When giving feedback, a reviewer (e.g., the instructor) clicks on the attainment level for each rubric. That creates a message to the student, listing what is right and

Exercise: Ordered list of dog breeds

Find a list of dog breeds on the Web. Create a Web page with a list of five of your favorites. List them in your order of preference: favorite first, next favorite second, etc.


Upload your page to your server, and add a link below.

Here's [my solution](#).

▼ Your solution

My solution is at <http://renata.com/coredogs/dog-breeds.html>.

Edit



Request review?

☐ No

Make public?

☐ No

May 31, 2012 - 9:21am
Page structure correct. Good code layout. Used the wrong tag for the list.
May 31, 2012 - 9:23am
Oh, I see. I changed from the ul tag to the ol tag.
May 31, 2012 - 9:24am
Good job!

Add

Figure 7. Exercise complete -

92 -

<p>Page structure</p> <p>Page structure correct.</p> <p>Page structure needs work.</p> <p>Page structure is way off. Use the basic Web page pattern from chapter 1.</p> <p>Code layout</p> <p>Good code layout.</p> <p>Code layout needs work.</p> <p>Poor code layout. Check our layout standards.</p> <p>Code layout</p> <p>Used the right lsit tag - ol.</p> <p>Used the wrong tag for the list.</p>
--

Figure 8. Clickable rubric

wrong with the solution. The reviewer can edit the text message before sending it.

Clickable rubrics let reviewers assess solutions to simple exercises in as little as 30 seconds. For the Web site dolfin, 30 - 60 minutes spent each day giving feedback is enough to keep up with a class of 40 or so.

DOLFINITY PRINCIPLE 5:

Dolphins have an efficient feedback workflow.

Most of the students' work is online, reading and doing exercises. However, individual contact is still important. Students meet with a tutor perhaps once per week, for one-on-one help with problem solving.

DOLFINITY PRINCIPLE 6:

There is one-on-one contact between students and an expert tutor.

There's much more to say about dolfinity. Some highlights: -

- -Dolphins are perfect for flipped classes.
- -Thousands of students can use the same dolphin simultaneously. However, students still receive individual attention through the feedback system, and in their tutorials.
- -Dolphins can be customized. Instructors can select content, choose exercises, and add their own material. Dolphins can even be branded for individual schools.
- -Exercise reviewers can be instructors, assistants, or other students.
- -Reviewers need not be local, or even in the same country.
- -Reviewers must have content skill, but need not be educational experts.
- -A student's solutions form a natural eportfolio.
- -Dolphins capture data about each student. If several exercises are due each week, the instructor knows which students are falling behind, without waiting for the midterm.
- -Dolphins can be updated instantly.
- -Student performance data can help improve dolphins. For instance, if students are consistently missing a particular rubric, then the explanation for that concept should be improved.
- -Students can work on dolphins at times and places that are convenient.
- -Since dolphins have many exercises, students spend more time studying than they may expect.
- -My experience is that teaching is more enjoyable when I drop lectures, and help students solve problems one-on-one.

Making dolphins

I envision individuals and small groups creating their own dolphins. Part of my work is making tools for them to do so. Some tools are ready, but much remains to be done.

Authors could offer dolphins for free, if they choose. However, authors should be able to charge for dolphins, under a business model that lets them keep most of gross sales. Authors could make a living from their dolphins, with the time and incentive to make them high-quality, up-to-date products. If authors receive the lion share of revenue, they could afford to sell access to their dolphins at a low price, and still support themselves.

This is, potentially, a new type of knowledge work. Academics from all fields could create dolphins, instead of or along with their university jobs. Dolphins could be hubs of intellectual exchange, as well as learning tools.

There are interesting scenarios. For example, two Detroit high school teachers might create math dolphins, specifically for that city. The dolphins would use local landmarks, like Ford Field. The exercises would include data from last night's Pistons's game, even relevant video from the game. The dolphins would help Detroit Public Schools implement flipped math classes, like those that have been so successful at Clintondale High School (Green, 2012). The teachers could sell their dolphins at a low price, and still support themselves. They could invest all of their work time in improving math education across the city.

There are many other possibilities, such as:

- Chemistry dolphins in Spanish for students in Florida.
- Algebra dolphins in Hindi for students in Pune, India.
- Writing dolphins for Oakland University students.
- (Your idea here.)

Conclusion

Dolfinity implements learning science principles to help students learn skills effectively and efficiently. Dolphins use people's time well. Human experts do the things that only humans can do: give feedback, and one-on-one help. Other things are left to computers.

You can see two dolphins at <http://coredogs.com>. I designed the tools used to make CoreDogs for my own use. They are not appropriate for, er, normal (i. e., nongeek) people. I'm working on new tools that are more usable.

The dolfinity business model lets authors make a living from high-quality dolphins. They do not need a publishing company. Small niches can be served economically.

Online learning is here to stay, but its final form is not yet clear. Now is the time to influence what online learning will become. It may be that dolfinity, or something like it, will be part of the future.

REFERENCES

Bergmann, Jonathan, and Sams, Aaron, *Flip Your Classroom*, ISTE, 2012.

Bransford, John, Brown, Ann, and Cocking, Rodney. *How people learn: Brain, mind, experience, and school*. National Academies Press, 2000.

Fagen, Adam P, Catherine H Crouch, and Eric Mazur. "Peer instruction: Results from a range of classrooms." *Physics Teacher* 40.4 (2002): 206-209.

Green, Greg, "My View: Flipped classrooms give every student a chance to succeed." CNN, May 31, 2012 <<http://schoolsofthought.blogs.cnn.com/2012/01/18/my-view-flipped-classrooms-give-every-student-a-change-to-succeed>>

Guskey, Thomas. *Implementing mastery learning*. Belmont, CA: Wadsworth, 1997.

Lester, James, Converse, Sharolyn, Kahler, Susan, Barlow, S.

Todd, Stone, Brian, Bhogal, Ravinder. "The persona effect: affective impact of animated pedagogical agents." *Proceedings of the SIGCHI conference on Human factors in computing systems* 27 Mar. 1997: 359-366.

Pears, Arnold, Seidman, Stephen, Malmi, Lauri, Mannila, Linda, Adams, Elizabeth, Bennedsen, Jens, Devlin, Marie, and Patterson, James. "A survey of literature on the teaching of introductory programming." *ACM SIGCSE Bulletin* 1 Dec. 2007: 204-223.

Robins, Anthony, Janet Rountree, and Nathan Rountree. "Learning and teaching programming: A review and discussion." *Computer Science Education* 13.2 (2003): 137-172.

Wiggins, Grant, and McTighe, Jay. *Understanding by Design, Expanded 2nd Edition*, 2005 Association for Supervision and Curriculum Development (ASCD), Alexandria, VA .