Captivate Article

Scenario-based learning in Adobe Captivate 2

Frank Nguyen

www.frankn.net

Table of Contents

Printable version

Send feedback

Get an e-mail update of new articles

Created:
        05 Oct 2006
User Level:
        Beginner, Intermediate

Although the training of software procedures is quite common, many developers are also asked to train employees on soft skills. Unlike software procedures that are more technical in nature, soft skills address personality and behavioral aspects of an employee. Common soft skill courses may include sales techniques, customer service, or management training One option to deliver these soft skill courses is to use an instructional method called scenario-based learning. The historical challenge with scenario-based learning through e-learning is that courses can be very difficult to develop using traditional authoring tools, due to the dynamic, branching character of scenario-based learning courses. Fortunately for our e-learning community, that is all changing. The same

evolution witnessed in software simulations is also happening to soft skill simulations.

Requirements

You must install the following software to follow the steps in this tutorial:

Adobe Captivate 2

Try   Buy

About the author

Frank Nguyen has managed the deployment of learning and performance systems for Fortune 100 companies. He is co-author of Efficiency in Learning (Jossey Bass, 2006) and has written numerous articles on instructional design and performance support. Frank holds an Educational Technology masters from Arizona State University and is a doctoral candidate focusing on the interaction of training and performance support.

Captivate Article

Scenario-based learning in Adobe Captivate 2

Table of Contents

The scenario

Imagine that you are working as an e-learning developer. You've been asked to create a course to train employees on compliance with Sarbanes Oxley (SOX) regulations. What would you do?

Let's say that you need to choose from one of the following options:

1. Develop a page-turner web-based training (WBT) course where learners read page after unending page of SOX rules and regulations. Just to keep people from dozing off, maybe you'll throw in a multiple-choice question here and there.
2. Develop a series of podcasts where C-level executives explain the importance of remaining SOX-compliant. To spice things up a bit, perhaps you'll do a vodcast (a video podcast) where learners can watch the CEOs head bobble to and fro while talking about lawsuits, bankruptcy, and doomsday.
3. Develop a simulation where the learners are immersed in an authentic business problem. They are provided with information and asked to make a series of decisions with the goal of remaining SOX-compliant while solving the problem. Based on their answers, the simulation branches into different paths in the learning environment, with realistic positive or negative outcomes.

It's not a scenario, it's reality

For many e-learning designers and developers, the dilemma described above is not an imaginary scenario. It is their reality. Although training procedures in software is quite common, many developers are also asked to train employees on soft skills. These needs can range from customer service, to people management, to regulatory compliance, to sales training.

So, which of the three options above did you choose?

For anyone that has chosen option one (the page-turning WBT) in the past, a quick tour of the company revealed a sea of learners halfway through the WBT lulled to sleep at their desks. The learners that were too impatient to endure the course simply skipped the content, went to the final test, and took it over and over until they reached the sacred passing score. While option two (a

series of podcasts and videocasts) can be a cheap way to deliver the message to the masses, the informal nature of this type of learning makes it impossible to track compliance, which is particularly important for initiatives like SOX. Since there are typically no smile sheets or tests associated with podcasting and informal learning, it is often difficult to measure the effectiveness of the intervention.

The third option (scenario-based learning), then, seems pretty attractive.

Captivate Article

Scenario-based learning in Adobe Captivate 2

Table of Contents

Scenario-based learning in Adobe Captivate 2

What option three describes is an instructional method known as scenario-based learning. In fact, the entire first section of this article is an example of scenario-based learning. Don't let the instructional design terminology intimidate you, though. There is no rocket science behind this idea. In general, you first immerse learners in a realistic scenario, provide them with a list of potential decisions, give them a description of the outcome of their choices, and then branch them into additional decisions and outcomes until they reach the end of the business problem. Figure 1 illustrates a scenario-based design for a lesson.

Legend

Preferred Path

Alternate Path
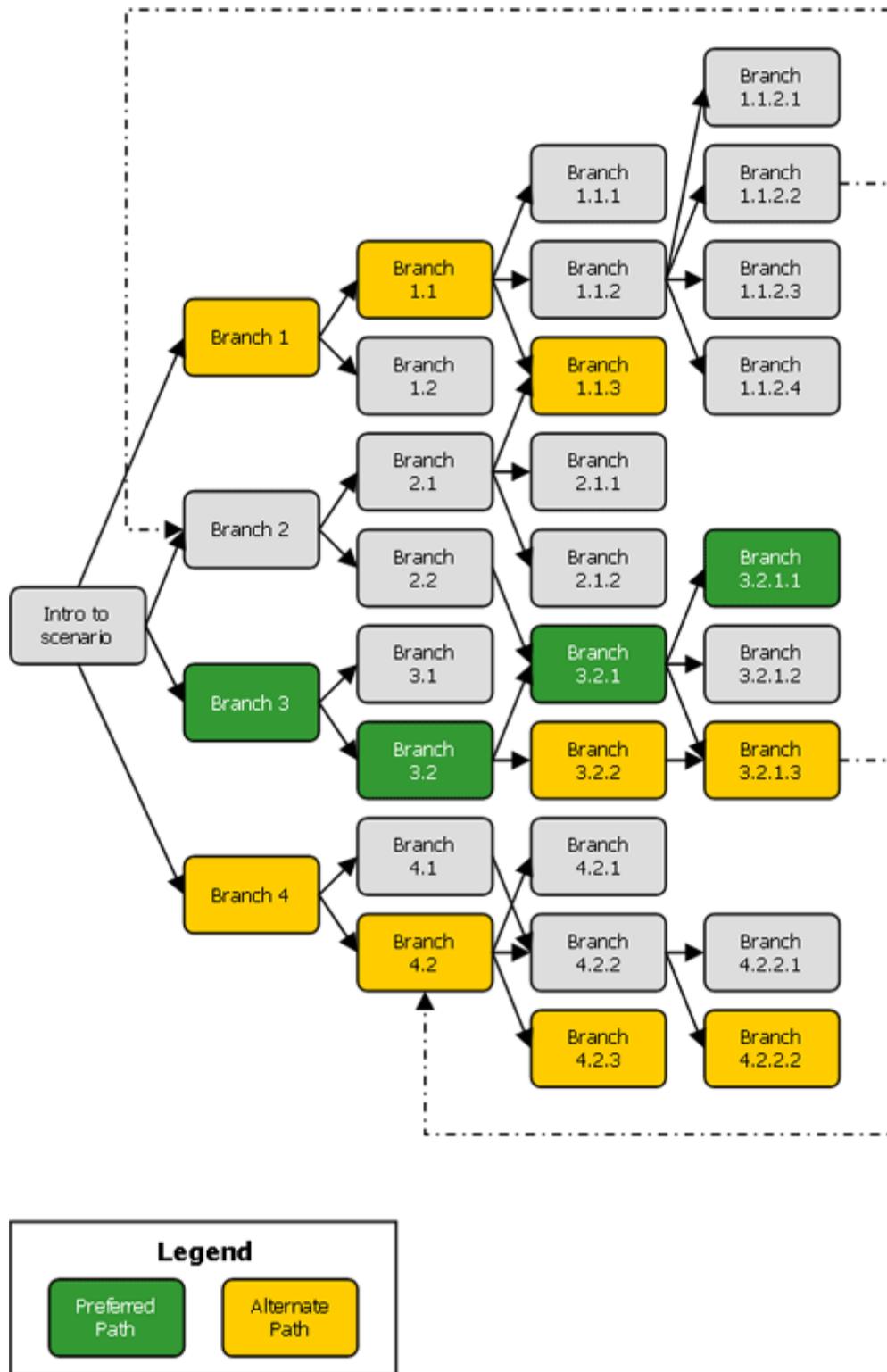
Figure 1. Example of a scenario-based learning design

In this example, each node in the scenario provides the learner with only three or four potential options to decide upon. When designing your scenario-based lesson, do not feel that you are required to provide every possible option. Doing so would be overwhelming to yourself as the designer or developer, as well as to the learner taking the course. Instead, audit subject matter

enough to find typical choices that employees are faced with in the workplace. Depending on the scenario, you may choose to provide the learner with as few as three choices or as many as seven. The options you present to the learner may comply with company policy, tribal knowledge or myths in your organization, or common mistakes a new employee might make.

Also notice in Figure 1 that the design incorporates a preferred path through the scenario. One or two alternate paths are also acceptable but not optimal. As you know, there is not always one right way to do things in real life, particularly with soft skills. Your scenario-based learning activities can be structured to reflect that. In addition to the preferred path, there will be numerous incorrect or non-preferable paths through the scenario. If the learner selects one of these paths through the scenario, you may choose to abruptly end the branch and provide them with timely and immediate feedback, perhaps in a funny or memorable way. By doing so, you can prevent the learner from continuing down an incorrect path for an extended period of time. It also removes unnecessary complexity in the scenario, and also removes frustration for both the author and learner.

The concept of scenario-based learning to simulate real life is not at all new. Malcolm Knowles (1980) pointed out that, unlike children, adult learners have "a reservoir of experience" which can be used as a "resource for learning." The instructional content you design should relate to the learner's prior knowledge and real life experiences. Proponents of authentic learning and constructivism will argue that knowledge is best acquired through working with and resolving problems in settings that approximate reality. This notion, from educational psychology, has been applied to numerous settings. Kindley (2002) simulated international crises to help decision makers deal with the complexities of global disasters. Colleges and universities have used the technique to build semester-long simulations to help expose students to the complexity of real-life problems (Ayers & Ostrander; Glasgow, 2006). Any designers who have also developed soft skill courses for the classroom have probably seen or used scenario-based learning activities in the past.

The historical challenge for scenario-based learning via e-learning is that scenario-based courses can be very difficult to develop with traditional authoring tools. You could use a traditional CBT authoring tool like Macromedia Authorware, but it might become difficult to build and manage all of the branches in the course. The same thing would happen with a presentation tool like Adobe Connect (formerly Macromedia Breeze) or software simulation tool like Macromedia Captivate 1.01. You could certainly develop the course using a more flexible tool like Macromedia Director or Macromedia Flash, but it would likely require someone with expertise in Lingo or ActionScript. Some companies have even resorted to building complex simulations in high-level programming languages like C#, Java, or VB.net, and have found them to be expensive to develop, maintain, and deploy.

**Rapid e-learning comes to scenario-based learning**

Fortunately for our e-learning community, that is all changing.

As part of a training organization that produces a large volume of software training, I have witnessed the evolution of software simulations over the last decade. I remember our first e-learning course in the late 1990's: A static screenshots of an application, linked images in an HTML document, and excruciatingly hand-coded HTML image maps to different parts of the image. (Those of you who remember Netscape Navigator Gold, Mosaic, or Lynx can commiserate with me.) You might spend fruitless weeks troubleshooting broken links. Anytime the software changed, you rebuilt the entire simulation. Mercifully, HTML editors, graphic editors, and eventually software simulations evolved to the point where you could create error-free simulations

with incredibly robust interactions in just a fraction of the time.

I believe the same type of evolution witnessed in software simulations is happening today with soft skill simulations. While many of us have been slaving away with complex HTML pages or Flash-based simulations, a number of soft skill simulation tools are now entering the market that will finally allow us to trade in the clumsy abacus for a fancy graphing calculator.

An example of this shift to rapid scenario-based e-learning is the new scenario simulation feature introduced in Adobe Captivate 2. When you open up Adobe Captivate, you'll notice that there is a New addition to the opening Project screen—a Scenario Simulation option.
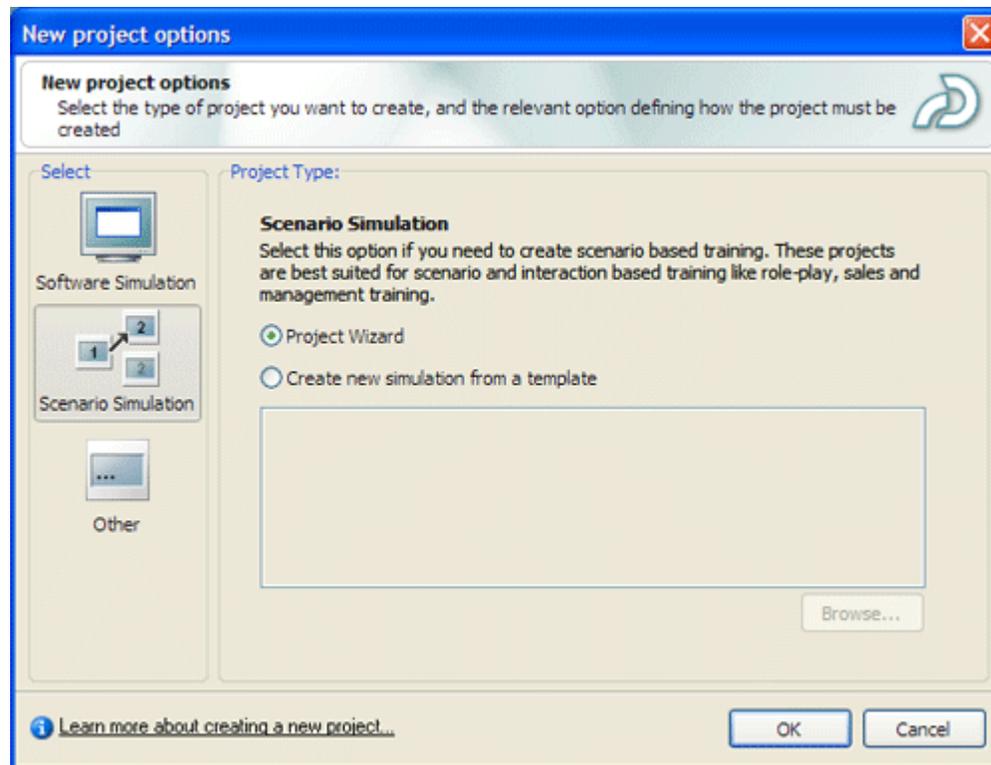

Figure 2. Adobe Captivate 2 contains new project options, such as Scenario Simulation

When you select the Project Wizard option, Adobe Captivate asks you to specify a number of familiar options, such as screen size settings, background color or image, and slide options. The most notable addition is the Scenario slides setting. As the user interface suggests, you can set this to the number of slides that you want in your scenario-based learning object. Don't get too hung up about getting the correct number right away, as you can easily add or delete slides later on.
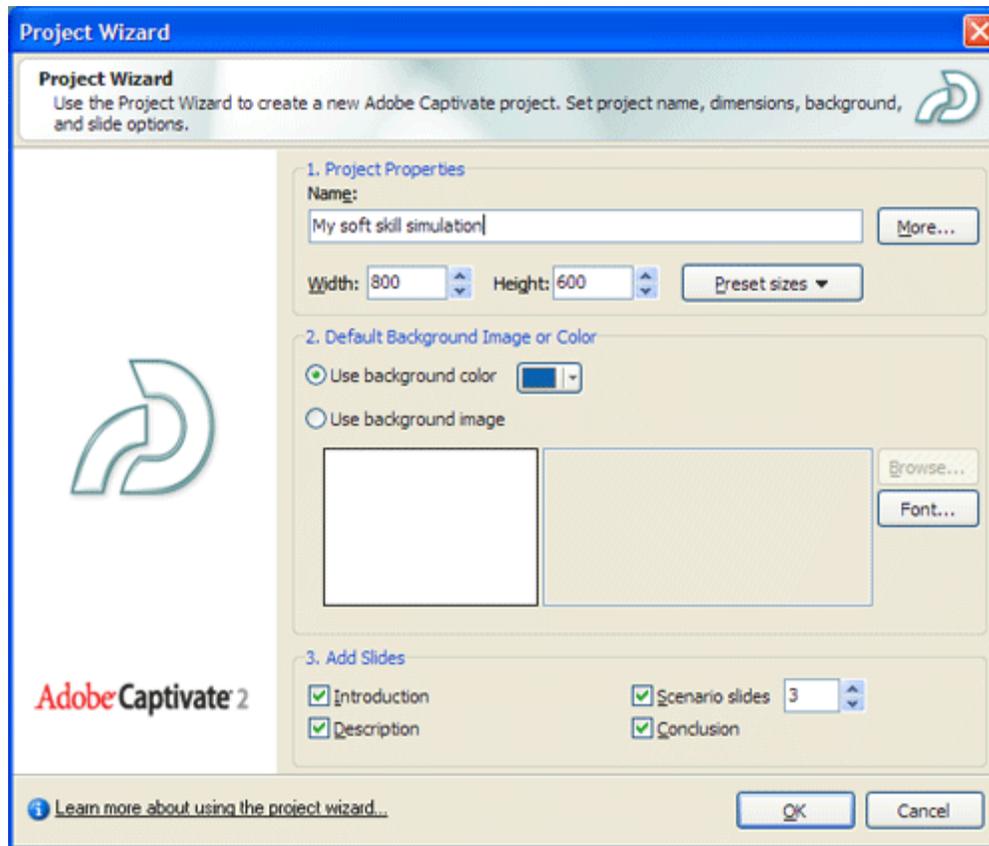
Figure 3. Adobe Captivate 2 scenario simulation wizard

The scenario simulation wizard generates a set of placeholder slides (empty slides you can fill in with content) based on settings you provided on the previous screen. Click ahead to the scenario slides. Notice that they look largely like standard Adobe Captivate test questions. To make the scenario more realistic, you can customize the question screens with text, graphics, or a video to introduce the scenario. Instead of multiple-choice answers, you can insert buttons with choices, which makes formatting the content easier than working with question slides. You could even have the user interact with an on-screen image where different parts of the images branch to different parts of the simulation. This type of branching may be particularly useful for non-soft skill scenarios involving physical tasks that you might find in a factory, warehouse, or even complex software simulation.
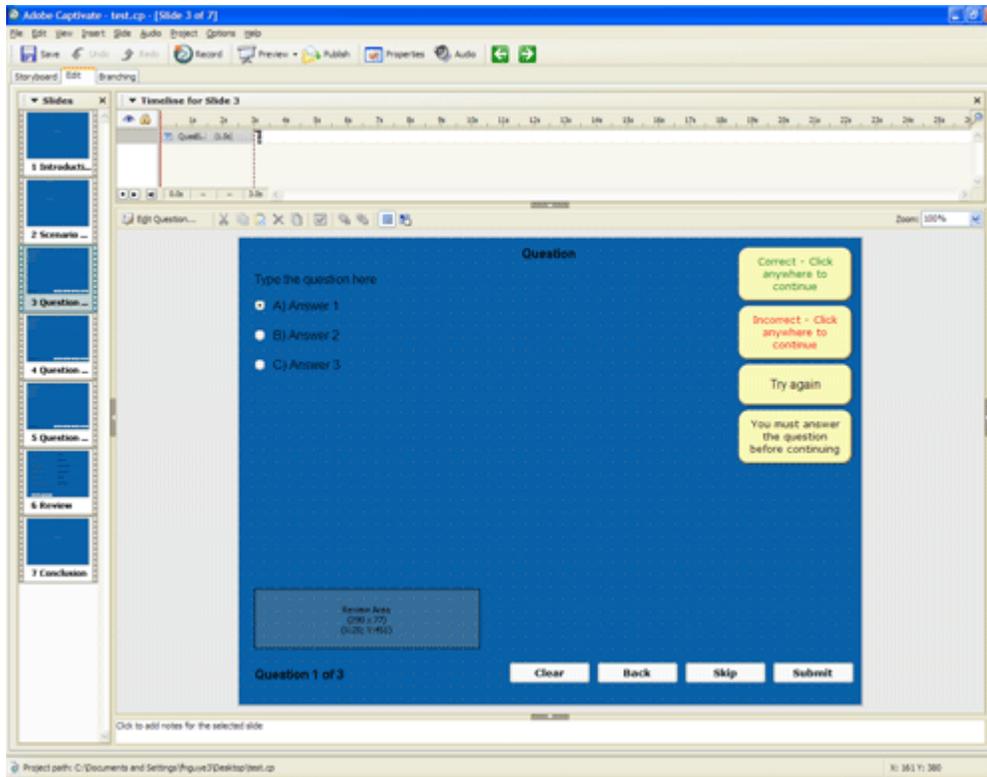
Figure 4. An example of a scenario slide generated by the scenario simulation wizard

As in Macromedia Captivate 1.01, when you edit a test question and go to the Options tab, you will notice that you have several actions you can set, such as whether the learner provides a correct or incorrect answer. You can still use this process to set up branching using Adobe Captivate 2.
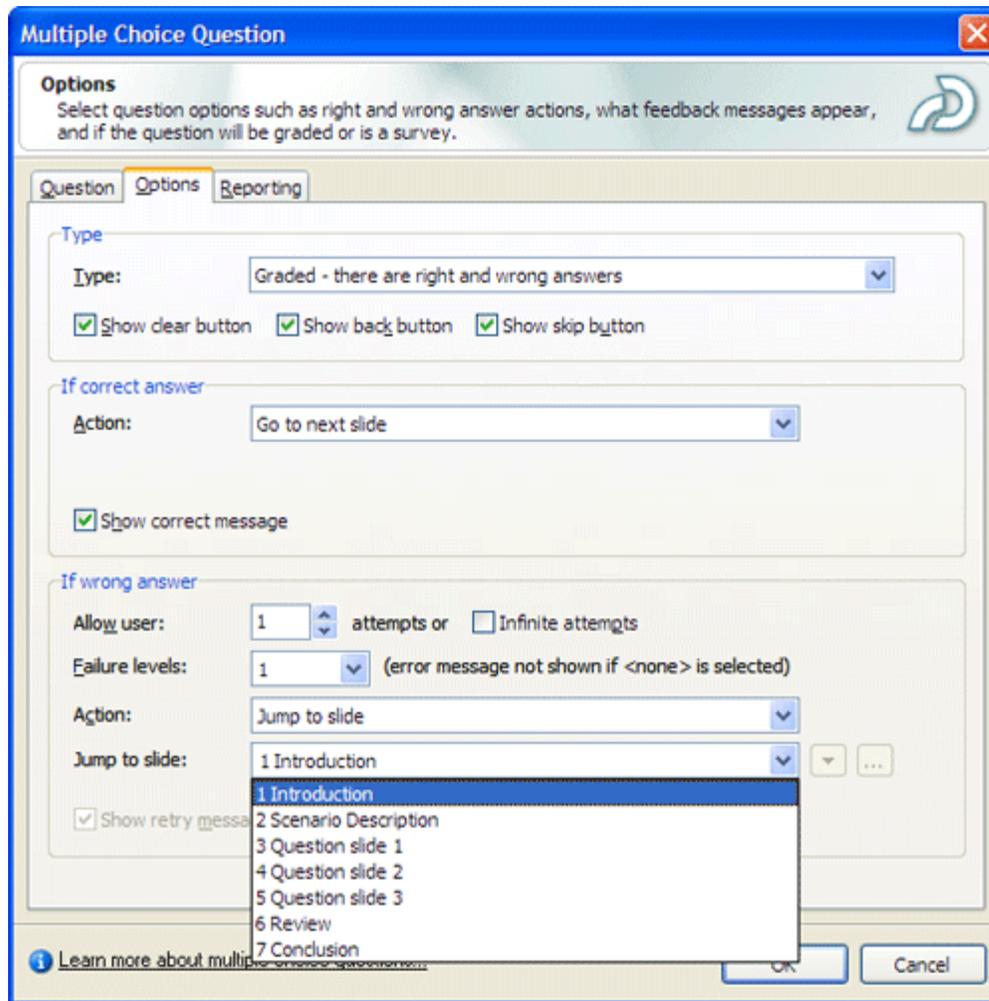
Figure 5. The Adobe Captivate 2 edit question window

But why do that when scenario-based learning is so much easier in Adobe Captivate 2, and its new Branching view feature. When you create a scenario slide, you can administer each path using the Properties dialog box on the right-hand side of the Branching view as shown below. If you create a scenario slide with more than two navigation paths (such as buttons or an image map), you can select each individual path, specify the action to be taken when the learner selects that particular path, and whether or not you want to track the path. You can expand and collapse parts of the simulation to simplify viewing. You can export the branching view to a bitmap or JPG to be viewed outside of Adobe Captivate. You can weight each path and send the data to your Learning Management System (LMS) using AICC and SCORM.
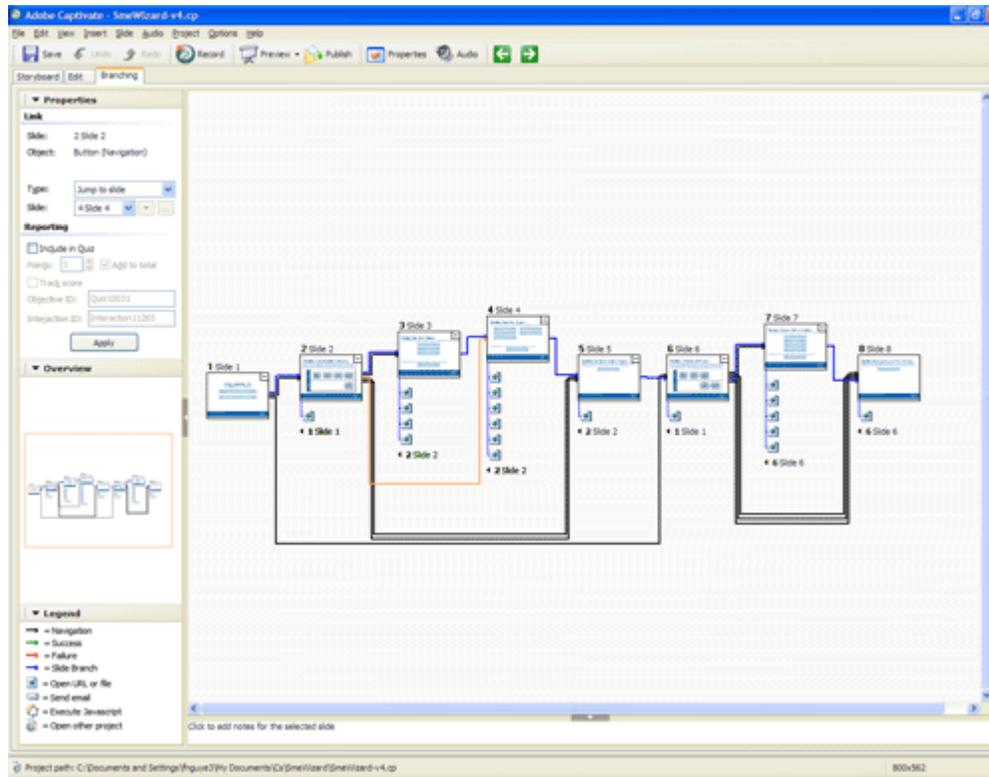
Figure 6. The Adobe Captivate 2 branching view window

Scenario-based learning is a powerful instructional method that designers and developers have long been able to leverage for delivery in the classroom. Thanks to the continuing evolution of authoring tools, the e-learning community finally has more powerful, easy ways to create them.

Where to go from here

To learn more about scenario-based learning, consult the online and printed resources listed in the references section of this document.

If you would like to learn more about creating scenario simulations using Adobe Captivate 2, visit.the Adobe Captivate Developer Center and Adobe Captivate Support Center.

References

Ayers, C., & Ostrander, J. (2005). Scenario-based learning for a more relevant student experience. Retrieved August 22, 2006 from http://www.league.org/publication/abstracts/learning/lelabs200512.html

Glasgow, J. (2005). Active learning strategies: Scenario-based learning. Retrieved August 22, 2006 from http://scope.citl.ohiou.edu/FLC/Active-learning/scenariobased.html

Kindley, R.W. (2002). Scenario-based e-Learning: A step beyond traditional e-Learning. Retrieved August 27, 2006 from http://www.learningcircuits.org/2002/may2002/kindley.html.

Knowles, M. S. (1980). The modern practice of adult education: Andragogy versus pedagogy. Englewood Cliffs: Prentice Hall/Cambridge.