chapter
SEVEN

Integrating and Piloting

We have discussed identifying the right place for highly engaging learning, gaining some support, and finally procuring a highly interactive virtual environment. The final items in our predeployment checklist are technical support for students, chunking, and piloting. (I will leave the other items of swearing, praying, and furrowing brow in confusion up to you.)

TECHNICAL SUPPORT FOR STUDENTS

After (or while) focusing on gaining access to the right HIVE, instructors also have to line up some technical support for students. Like a napkin at an outdoor picnic, some might optimistically think they don't need it, but all will and you don't want to be caught without it.

Support can be anything from providing help with logins and passwords to making sure browsers are up to date and software is installed correctly. Support can be needed because the computer doesn't have the necessary capability, there are problems logging on, the right browser isn't installed, or an update for Flash or DirectX is missing. Some hard-to-diagnose problems can occur if a student is running a processor-intensive program in the background, like a 3-D golf game.

Support is especially tricky because professors are often less technically literate than their students. As with instructor manuals, the role of technical support can come from many sources. One solution is to set up or use a third-party support system, as is often provided by vendors. But if the simulation is free or heavily discounted, the support is inevitably lacking.

If the professor is indeed the point person, he or she must know how to access the software and be aware of the two or three most likely problems. Where possible, a teaching assistant who is technically literate and has more free time can

take on this troubleshooting role. He or she can be directly connected to a support community, formal or informal. If a TA is the tech support, make sure his or her name is well distributed to all students.

Some organizations have computer technical support groups that can take over this burden as well. They just need to be briefed before they are called. They hate surprise calls. Really.

On top of the need to line up a troubleshooter, the reality of technical glitches makes it absolutely critical that students install or access any software at least twenty-four hours before they need to use it. For example, if students will be using an environment for the first time in the classroom, it may make sense to schedule an installation or logon event at the end of the class immediately before it.

CHUNKING CONTENT

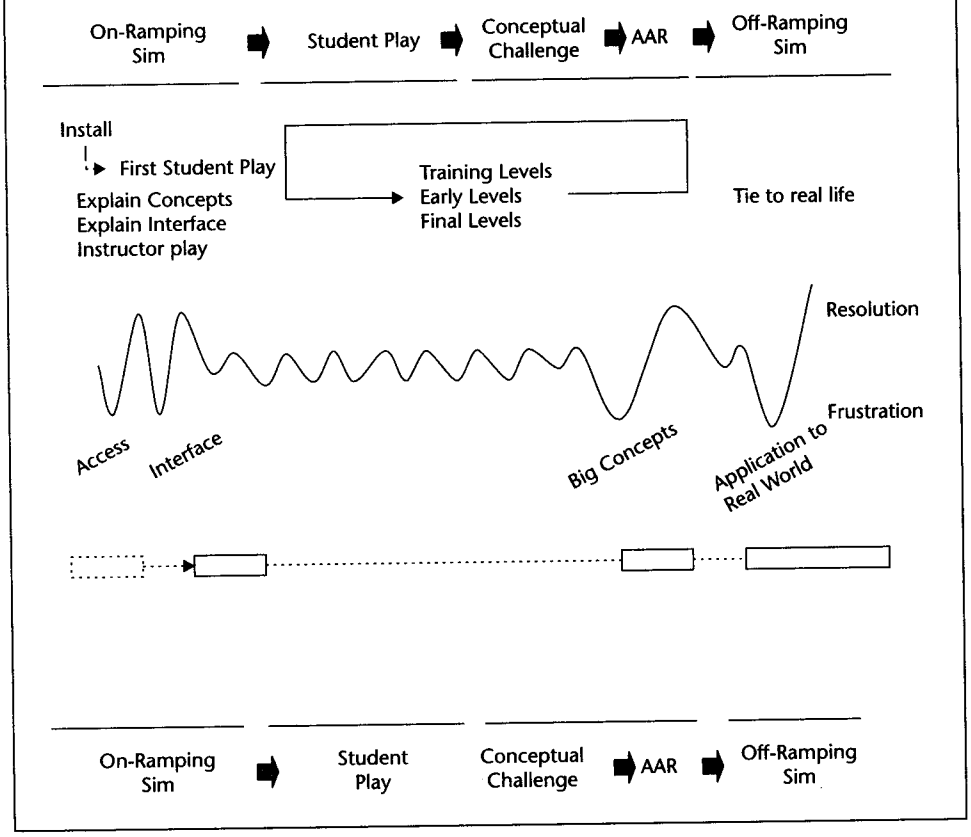
Any long interactive content, such as a stand-alone sim on leadership or a group activity in *Second Life* to build a giant model of a cell, has to be integrated with the curriculum. Instructors have to decide how many days will be spent on the activity, and for those virtual classes that have them, how much of the simulation will be spent during "live" (synchronous) class time versus chat rooms versus office hours versus homework, and how the simulation itself will be punctuated by classrooms.

As a simple example, imagine a single-player asynchronous simulation, with no built-in synchronous pieces, that takes fifteen hours to engage. Imagine now that it is to be deployed in a classroom that meets for one 120-minute session once a week. This brings up quite a few possibilities and permutations. Should the entire simulation be played between two synchronous classes? Or should students engage half of the simulation, then attend a class, and then play the second half of the simulation? How should "live" synchronous events, if they are to be included, be allocated and best used?

Students in simulations go through cycles of frustration and resolution (as we discussed starting on page 40, and which we will talk a bit more about these later). In the "lows," students are most prone to quitting the sim, but they are also on the verge of coming to a resolution, which results in new mental muscles being formed (see the squiggly line on Figure 7.1).

The waves of highs and lows can be fairly predictable, especially on the third or fourth iteration of using a sim. Lows typically come around installing, accessing,

Figure 7.1
The Process of Using a Simulation



or teleporting to the sim for the first time, around using the interface, around every major new concept, and often when the student is challenged to create a strategy to apply the content to real life.

Thus, synchronous events such as classrooms (face to face or virtual, represented as the rectangles in Figure 7.1) should be used first to launch the simulation, then a little bit after each major frustration point as an After Action Review, and finally to tie the sim back to real life. In completely asynchronous environments, instructors may have to use office hours or proactively reach out to students via e-mail in anticipation of these lows.

Collaborative or team-based sims, in contrast to the single-player scenario, offer a further level of chunking requirements, often by the students in off-class

(i.e., homework or lab) time. Suppose a sim on running a hospital is made up of ten turns, with each turn requiring team collaboration, and the entire sim has to be completed in two weeks. Instructors may assign one turn a day for five days a week. However, this requires a significant "live" commitment from the students (which is one reason the University of Phoenix minimizes use of this model and prefers the more self-paced branching models).

Some sims end with a big aha or gotcha (a lesson that if known about beforehand could subvert the learning), in contrast to sims aimed at practice. To prevent cheating or other bypassing of the learning goals, their content should be engaged in a proctored and synchronous environment.

Using Games at Points of Stress

There is another way, perhaps a better way, to deal with the moments of high frustration. Class games and activities, typically between five and twenty minutes long, are often a better option either right before or during moments of high frustration to both release stress and get to resolution more quickly.

Many class games are specifically aimed at reducing the frustration of learning an interface. For example, to teach students how to use the building authoring tools, they may be given the task of creating their own version of an existing object in *Second Life.* Or students may be asked to draw a face on the screen of a virtual classroom session to learn how to use the markup tools. Or to e-mail a screenshot as a postcard. To make the task more fun, students may be broken into teams and given time limits.

Later on, students may be given a brainstorming game to identify terms they have learned so far in a word jumble. Or to compete in a simplified version of the sim.

Printable Guides and Workbooks in Simulation Deployment: Friend or Foe?

Finally, the instructor has to think about support material. At the very least, a good downloadable tip sheet can save a bit of aggravation. But what about a workbook or guide? That's a good question.

Printed workbooks and guides are so, well, 1990s. They are packages of, gasp, paper (even if printed from PDF originals). They have the old-fashioned inputs like check boxes and blank places to write free-form text.

Their pages are bound together, forcing linear access. They cannot be graded electronically. If you find a mistake in a published guide, you can't correct it instantly.

In contrast, when one thinks of simulation deployments, one thinks of completely online experiences. Participants might get a CD-ROM, DVD-ROM, or ideally just a link. The experience should be futuristic. Any help is just-in-time and on the screen.

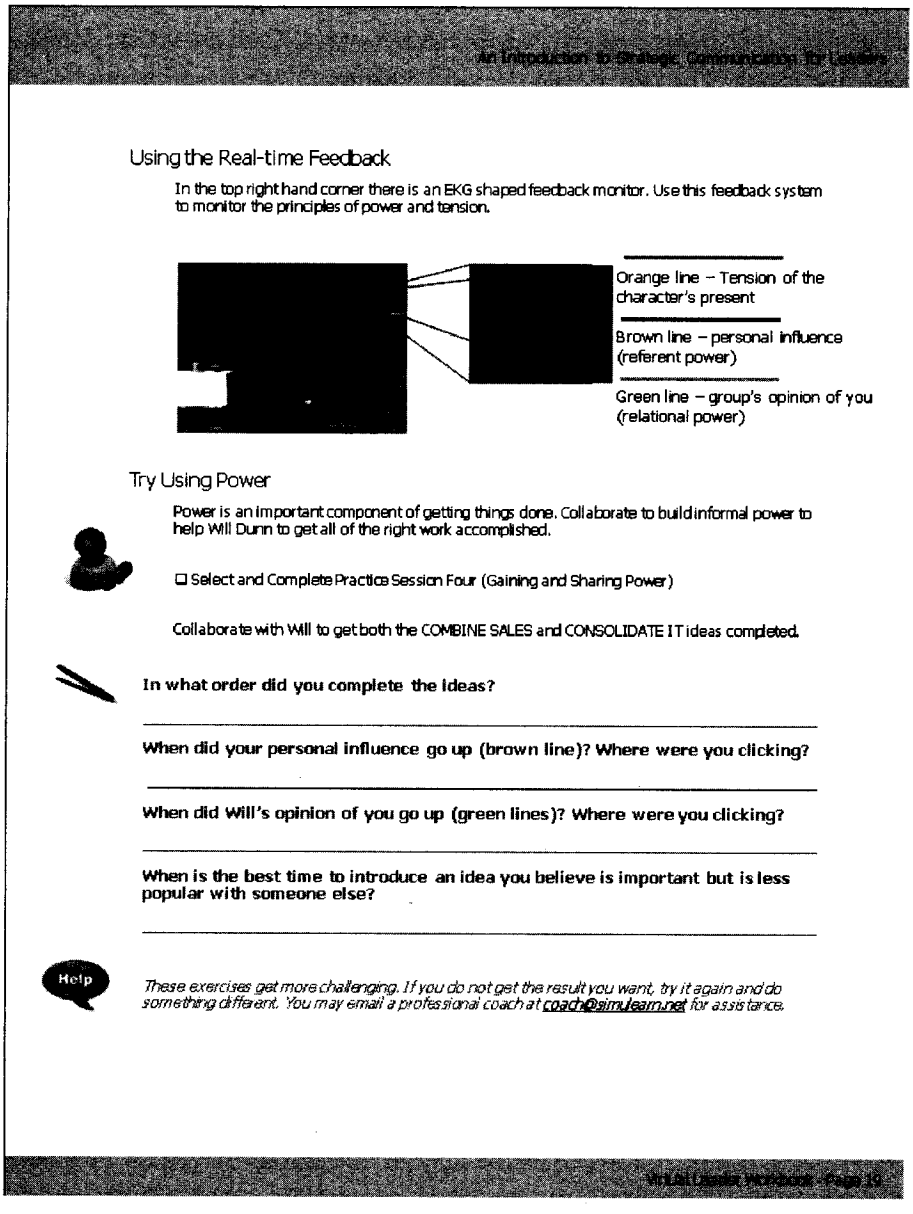
So why do I have the nagging feeling that workbooks and work guides will play an integral part in simulation deployments in the years to come? I guess because they have so many advantages. It is hard to break out of the simulation once you are in it. The very immersiveness of the simulation can almost trap players. Therefore, having a parallel content source, providing help, asking questions, guiding thoughts, and giving tips makes a huge amount of sense. In addition, printed guides are familiar. They are paper-based. They are comfortable. In the potential chaos of the simulation and other electronic environments, the paper can almost be reassuring.

Workbooks and guides provides a lasting artifact. Even after the link is lost and the simulation put away, the pages provide immediate reminder of the activities. And workbooks are easy and cheap to update, especially for a professor. Workbooks are often created in PowerPoint or Word. To change a term, a definition, even lesson plans, it can be much easier to do it in the paper world than in the electronic world. A single workbook can also tie together three or four disparate simulations.

The theorists will probably suggest that most if not all positive attributes of printed workbooks or guides, including those listed here, can ultimately be subsumed in the simulation itself, including support pages before and after any core gameplay. And I can't disagree. But my practical side has a sneaking suspicion that workbooks used in conjunction with simulations will be around for quite some time.

Perhaps the killer app in support of workbooks is that, increasingly, professors will access sims the way they access textbooks—as off-the-shelf finished products. The only way, or at least the best way, to customize the experience will be through other artifacts. In some cases professors may go through the basic play themselves. For example, they may take the entire class through the first level. Whether the guide is a one-page handout or a full tome, decide on your own level of need and desire. Then if you need one, build it.

Figure 7.2
A Page from a Workbook Supporting SimuLearn's vLeader



An Introduction to Strategic Communication for Leaders

Using the Real-time Feedback

In the top right hand corner there is an EKG shaped feedback monitor. Use this feedback system to monitor the principles of power and tension.

Orange line – Tension of the character's present

Brown line – personal influence (referent power)

Green line – group's opinion of you (relational power)

Try Using Power

Power is an important component of getting things done. Collaborate to build informal power to help Will Dunn to get all of the right work accomplished.

☐ Select and Complete Practice Session Four (Gaining and Sharing Power)

Collaborate with Will to get both the COMBINE SALES and CONSOLIDATE IT ideas completed.

In what order did you complete the ideas?

When did your personal influence go up (brown line)? Where were you clicking?

When did Will's opinion of you go up (green lines)? Where were you clicking?

When is the best time to introduce an idea you believe is important but is less popular with someone else?

Help

These exercises get more challenging. If you do not get the result you want, try it again and do something different. You may email a professional coach at coach@simulearn.net for assistance.

vLEAD Leader Workbook - Page 10

PILOTING

The final part of prep work is piloting. Highly interactive virtual environments need to be piloted. They need to be tested to hone the process of deployment. In fact, they need to be piloted twice. They need to be tested first to make sure all the technical bugs and processes are smoothed out. And then they need to be piloted conceptually with a class to make sure they do what we say they can do.

Technical Pilots

Let me be more specific. A technical and process pilot will happen regardless of what you do. The only question is, do you want to pilot with a real class, or a test group?

Technical pilots should, as close as possible, resemble the real-world environments in which the simulations will be deployed. It is not enough, for example, to test a deployment that will be international by testing it in different locations in the same building.

Different browsers should be used in the technical pilot if different browsers will be used by the students. If the simulation requires the use of a network, make sure that the different firewalls, including corporate if some students access from their workplaces, will accept the network stream.

Stress testing is also critical. If 30 or 50 or 1,000 students will be accessing the content simultaneously, those are the conditions that need to be tested. However, stress testing servers in one environment before testing them in a distributed environment allows faster troubleshooting.

Also in doing a pilot, especially if using Web-based content, beware of *caching*. Caching is the process by which a local computer temporarily stores content that it pulls from the Internet. The reason caching can be tricky is as follows. The first time content is accessed remotely, the videos play choppily. The technical person makes a few adjustments and thinks she has solved the problem. The remote test user plays the video a second time and it runs smooth as butter. Ostensibly, it seems as if the problem is solved. In fact, the only reason the performance has improved is that the video has been cached in the local computer. When a student loads it for the first time, he or she will still have the choppiness problems. It is often worth running a free tool such as CCleaner (www.ccleaner.com) that cleans out all cookies and cached content before any final technical test implementation.

Making Sure the Communication Infrastructure Works

The next vital activity on the checklist is to make sure the communication infrastructures are working. Technologies that have to be considered for any good collaborative experience include

- message board/forums and chat rooms,
- instant messaging,
- application sharing (make sure screen-sharing technology includes the mouse position, which is critical for live demonstrations),
- separate virtual rooms for teams to break off into,
- telephone conference call number,
- voice-over IP,
- calendaring tools, and even, where appropriate,
- multiplayer mediation.

Some environments have their own built-in communication infrastructure, such as their own version of instant messaging. But where possible, use instead the existing infrastructure students already use, including the tools included in the learning or class management system.

An in any kind of team-based project, a wise instructor will allow for both one-to-all communication and one-to-one communication. This is critical. Harvard Business School Publishing's Denis Saulnier explains why:

> If a team leader turns out to be overbearing, we will see individual team members chatting with each other about that behind the leader's back. That becomes an important dynamic in terms of how the simulation is played. Now whether or not to save those side conversations and make them available to the instructor has become a point of great interest. Currently we do not do that. Most of the faculty members believed that students had to feel as if that communication was truly private, even in a distributed environment. But others are trying to figure out some way of capturing that information, and we may just do it in an anonymous way, or perhaps aggregated way, at some point the future.

A Harvard Business School Publishing/Boston College distributed-learning pilot used both Web chat and speakerphone. Students very quickly learned to

comfortably use both media. Typically, comments for the entire group were spoken and comments to individual team members were typed.

Conceptual Pilots

Although the technical pilot is critical, the pilot that more people will care about is the conceptual pilot. This piloting is the last step before a major rollout, with either a 10 percent or a 30-person sample of representative participant members (whichever is smaller), running the program exactly as you intend the full program to be but measuring the results at the end more rigorously.

The process of the conceptual pilot will match very closely, hopefully identically, the final rollout, as described in the next section on simulation deployments. Use that section to guide your conceptual pilot process. When HIVEs are short, such as ideas for new class games, they can also be tested in a pre-class session. If the students are asked to log in ten minutes before the class starts, they can be used as a test audience for a later class.

The results of a pilot can be used to make a "go/no go" decision if an organization is evaluating a sim or to practice and calibrate the sim before the full rollout (these are very different goals, and they do affect the feel of the conceptual pilot if not the steps). If the sim is being calibrated and significant changes are made, another pilot deployment is suggested before a major rollout.

As a quick note, when piloting a sim, it is critical not to say to the participants, "We are piloting this program. What do you think?" Instead, it should be positioned similarly to the major rollout, such as "This is the required program. Do your best."

CONCLUSION: THE NEED FOR FRONT LOADING

There are a lot of steps in preparing. It seems like a lot of work, and it is. But there are two comforting pieces of news: The first is that by preparing well, you will save yourself a lot of headaches later on. The second piece of good news is that once you have done this, in most cases you won't have to do it again for the same sim.

Like parenting, simulation deployment tends to be front loaded. This is a theme we will come back to in Chapter 9. Some instructors will even feel guilty later on when their students are working hard and they have nothing to do.