DESIGNING EFFECTIVE E-LEARNING USER INTERFACES

David Guralnick, Ph.D.
Kaleidoscope Learning
304 Park Avenue South, 11th Floor
New York, NY 10010
dguralnick@kaleidolearning.com

ABSTRACT

The user interface is, of course, a crucial part of a user's experience with any piece of software. For e-learning sites, interface design is particularly critical, as the learning effectiveness and interface design are substantially intertwined. The focus on the need for highly-usable interfaces in corporate e-learning has had an unexpected negative effect on the quality of e-learning sites: in their zeal to ensure that learners are never confused by an interface, designers have tended to design simplistic products that do not result in the learners actually learning the needed skills. I suggest in this paper that e-learning sites can involve complex tasks for the learner yet still be intuitive and easy to use, if designers follow an interface design process that is determined by how people learn and the tasks they need to perform in the program. This is in contrast to other approaches which view the interface design process as separate from the learning design. In this paper I describe an approach to e-learning interface design that focuses on the goals, audience, and learning, and explore examples that demonstrate the success of this approach.

1. INTRODUCTION

The user interface is, of course, a crucial part of a user's experience with any piece of software. E-learning interface design is especially critical, as the learning effectiveness and interface design are substantially intertwined. In addition, a trend toward ensuring the usability of e-learning sites, particularly among large corporations in the U.S., has led to a "usability-first" way of thinking about e-learning design, in which ease of use is considered to be the top design priority. While on the surface this argument seems valid, such an approach is actually in opposition to sound design principles—a designer following such a method becomes focused primarily on making a product easy to use and in doing so, does not focus sufficiently on whether people learn from it and are motivated to use it.

I suggest in this paper that e-learning interface design should be a core, integrated component of the overall design of an e-learning product, and that interface design should be determined by how people learn and the tasks they need to perform in the program. This is in contrast to other approaches which view the interface design process as separate from the learning design, often led by a graphic designer with no specialized knowledge of or experience in learning theory. In my view, the right question for an e-learning designer to ask is "what should the learner be able to do?", rather than "how can I make it clear what the learner should do next?"

In this paper I describe an approach to e-learning interface design that focuses on the goals, audience, and learning, and explore examples that demonstrate the success of this approach.

2. COMMON MYTHS ABOUT INTERFACE DESIGN

The role of "interface designer" has become more and more widespread, particularly since the invention of the World Wide Web and the resultant explosion of interface designers and developers. E-learning, as a still-emerging discipline of its own, has suffered from a dearth of trained, experienced e-learning designers, particularly in regard to corporate e-learning programs in the United States. Perhaps due to the designers' inexperience, several common opinions about interface design have emerged, as I have found in my consulting work and in my speaking engagements (Guralnick, 2006), with which I disagree. These include the following:

- *Myth*: Interface design is about "navigation". *Reality*: A learner using an e-learning program is trying to do something, not simply to go somewhere for the sake of going—thus the designer's focus needs to be on putting the learner in an appropriate situation to learn, and creating an interface that supports the actions the learner can take. The end result should be an interface that is intuitive to use (though "use" may, and often should, involve a lot more than just "navigating," or moving around the program), but the top-level goal of interface design should not be to be navigable.

- *Myth*: If the user always knows what to do next in a program, so therefore the program is "usable," then the program is a success. *Reality*: Certainly it is true that if the program is unusable, it is not a success, at least in the immediate sense of being useful (it may make contributions to future works, even as a failure). And some excellent work has been done in software usability, most notably by Nielsen (Nielsen and Molich, 1990; Nielsen, 1993). Yet in some ways Nielsen's concepts have been frequently misapplied; the goal of an e-learning program is for people to learn; in the corporate world, the goal is specifically to improve the learner's job performance. An unusable product will prevent the goal from being achieved, but usability is not the sole goal in itself, and it is very simple to build overly simplistic programs which are "usable" at all stages (e.g., the learner must click the Next button) but do not teach anything.

- *Myth*: Most of the work involved in interface design requires expertise in video and Flash. *Reality*: While there can be many useful roles for video and Flash animations, I suggest that both should generally be used only to serve a learning purpose (including motivation, however). For example, a program that teaches coaching skills by asking learners to analyze video clips of coaching situations makes excellent use of video for learning purposes.

- *Myth*: Graphical art skills are necessary to be an interface designer. *Reality*: While graphics skills are needed to create a nice-looking interface, I suggest here that the graphics role in an e-learning project is somewhat analogous to the role of a construction worker implementing the design of an architect. The design architect, under this set of roles, is responsible for determining the learner's experience, and generally should understand learning theory as well as interface design, but does not need to be able to create the graphics himself.

In the remainder of this paper, I will describe a program, the Directory Assistance Training Course, which has a very successful, yet somewhat complex, interface; then I will describe a generalized process which can be used to produce effective and engaging interfaces, annotated with examples from the creation of the Directory Assistance Training Course.

3. AN EXAMPLE: A COMPLEX YET SUCCESSFUL E-LEARNING INTERFACE

3.1 Overview

Ameritech (now part of SBC Communications), the telephone company in the midwestern United States was looking to improve the performance of their directory assistance operators. They needed a more cost-effective way than their existing classroom training course to teach operators to handle directory assistance calls quickly (keep call times short) and accurately (provide the correct listing information). I led a team that was brought in to develop what would become the computer-based Directory Assistance Training Course.

Prior to the existence of the computer-based course, operators were hired and put through a weeklong classroom training session, which mainly focused on using Ameritech's specialized computer hardware and software to look up phone listings efficiently. Becoming an expert directory assistance operator is much more difficult than it first appears; operators need to become skilled in employing the optimal "keying strategy" in a variety of situations. For example, when a caller requests a phone listing for a residence (rather than a business or government), there is a particular rule to follow: the operator should enter the first five letters of the requested person's last name, followed by a ".", followed by the first letter of the person's first name, followed by another ".". The "." characters serve as wild-card characters, and each rule has a series of exception cases (e.g., suppose the requested person's last name has only four letters, or their first name is not known by the caller) which the operators must learn as well.

The computer-based course focuses on teaching new operators (and sometimes experienced operators, as refresher training) how to efficiently find phone listings. Operators in the course learn by taking practice "calls" in a simulated computer environment. The "calls" are in audio, and the operator uses the same keyboard and screen in the training as in the real job. In addition, the screen in the training system is augmented with a few other features, most notably a guidance and feedback "tutoring" component to help the operator through the listing-lookup process successfully. The entire course contains fifty simulated calls, or scenarios, and takes approximately four hours to complete. The overall training course, with the simulation included as a key component (the rest was classroom-based and focused more on customer service skills than keying), was condensed from five days to two days total.

In the remainder of this section, I will describe the Directory Assistance Training Course in more detail; it provides an example of a course with a complex yet very successful user interface, and the process used to design this interface can be nicely generalized to be used for other programs.

3.2 Sample Screens and Learner Interaction

In Figure 1, we see the learner's screen at the beginning of a simulated call, in this case a request for a fairly simple listing, the Treasure Island grocery store in the town of Wilmette.

Figure 1. The learner's screen, after a call has come in.

At this point (and at virtually any point in the simulation), the learner can do one of several things:
- Ask a question of the tutor (Now What?, How?, or Why?);

- Choose to speak to the customer via the Talk to the Customer button, which in turn brings up a set of options;
- Look for general assistance (e.g., a Tour of the Operator Screen);
- Use his keyboard to enter information.

Figure 2 shows what happens when the learner (a new operator, who has used the tutoring component for help) has successfully entered in the "location" information for the town of Wilmette, and has then asked the tutor "Now What?"—that is, "what should I do now that I am finished with the location?" In Figure 2, we see the tutor's response and the current state of the learner's screen.

Figure 2. The learner's screen, after asking Now What?

After the learner successfully enters the business name, he then must press the Bus (for "business") key on his keyboard, and then gets the listing as shown in Figure 3. Sometimes the listings are exact, sometimes the listing screen will show many listings. Finally, the learner should press the "Audio Rls" key on his keyboard to play an automated recording of the phone number for the customer. At that point, the simulated call is considered to be complete and the learner moves on (actually is moved on, the call simply comes in over the learner's headset) to the next.
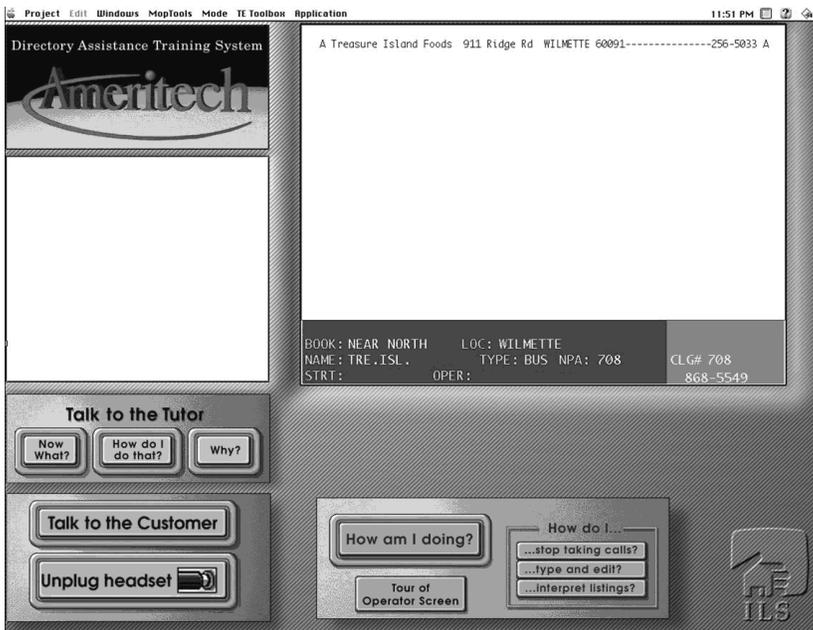
Figure 3. A listing screen.

Figure 4 shows one additional feature, the on-screen keyboard that flashes a key when requested; if the learner asks "Now What?" and is told to press a key, and then asks "How Do I Do That?", he is shown the keyboard in Figure 4 with a key "flashing" to direct his attention to where it is. The on-screen keyboard exactly mirrors the learner's keyboard.
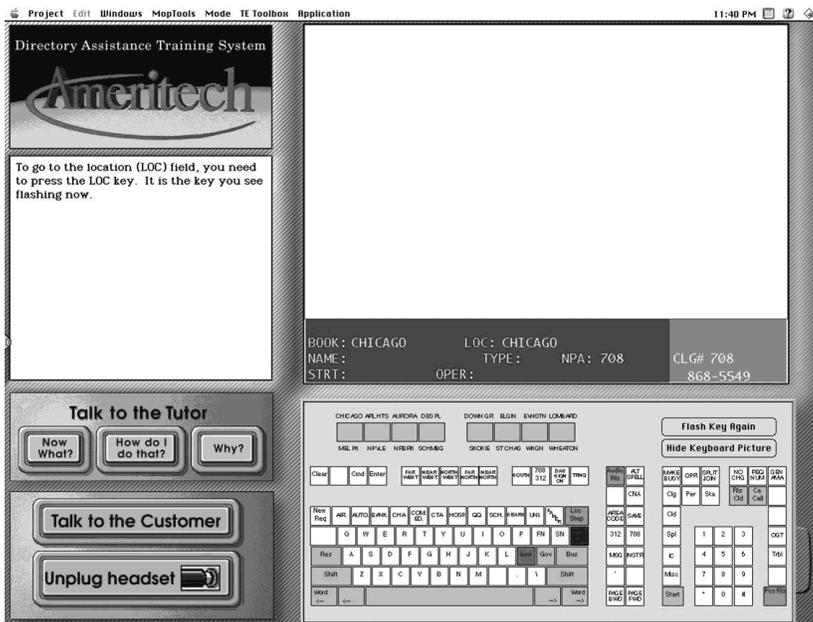

Figure 4. Showing the learner where the "LOC" key is.

3.3 Summary of the Learner's Experience

The learner begins the program with very little background information, and is told by the program (in text) that he will take simulated calls, will be measured on speed and accuracy, will have a tutor to help him, and that it should not be concerning if the calls seem difficult at first, it takes time and practice to become an efficient operator. After the brief introduction, the learner is walked through a short login process and then told to ask the tutor as he wishes, then is thrown immediately into the simulation and starts receiving simulated calls. The calls begin with the easier situations (no special cases) and gradually become more complex over the fifty call scenarios. The learner's performance along speed (time per call) and accuracy (number of keying errors) is tracked and is available for learners to review between simulated calls.

3.4 Results

Given that the learner's task is, while not tremendously complex, not obvious (there is no way to guess the keying strategies, and nothing particularly intuitive about them), and that a fair amount can happen on the learner's screen, one might have wondered if the Directory Assistance Training System's interface would confuse some learners. At a glance, it may seem to be a lot to understand without much background information, especially for an audience with little computer experience.

In practice, however, the results were excellent. A pilot version of the program garnered extremely high usability scores when tested by Ameritech's human factors professionals, and the anecdotal feedback from learners, once the course was rolled out, was that the course was incredibly easy to use. A performance evaluation by Ameritech then studied two groups of new operators, one who took the old, entirely-classroom-based course, and one who took the new course focused around the simulations. The two groups were found to be similar in their accuracy ratings, but the computer-based group far outperformed the classroom-based group when it came to speed, clocking in at one and a half seconds faster on average. Ameritech's wanted its operators to average below twenty seconds per call; a 1.5-second improvement was considered significant and resulted in a savings of over $5 million annually.

4. DIRECTORY ASSISTANCE TRAINING: WHY IT WORKED

Below I list some reasons why Directory Assistance Training was considered so successful—even with what could be termed a "complex" interface and fairly novice computer users, learners felt comfortable with the program. These reasons were derived partly on informal interviews with learners. The primary successful features of the program and its interface included:

- The interface actions supported the tasks the learner needed to perform;
- The screen layout drew the learner's eye to the right things (for example, the eye would focus on the white screen area, but the tutoring buttons are also easy to see and read);
- Less-crucial actions (e.g., Talk to the Customer options) were hidden rather than cluttering the screen;
- A tutoring component was always available to the learner;
- The entire program emphasized in-context, just-in-time learning over general information;
- The feedback and guidance were designed and written to aid generalization—e.g., showing a rule plus an example;
- The program design cut off learner paths so the learner stayed goal-focused; yet taught error recovery where it would be needed in real life (e.g., if the learner moved his cursor to the wrong field, he needed to learn how to get back to the right field);
- Learners did not need to click the mouse more than twice to complete an action;
- The program was engaging because the task was challenging and realistic, and because the performance measures (speed and accuracy) were the same measures used on the job.

The reasons above are results, in that they are descriptors of the interface which seem to have been successful, and may well be worth replicating. But then perhaps the larger question is how to get to this point—that is, what type of design process could yield an interface that is successful. The argument here is that the features alone should not be replicated, but rather a design process that led to these successful features should be replicated—each project is different, with its unique characteristics, and successful design is specific rather than boilerplate. In section 5 below, I lay out such a process.

5. E-LEARNING INTERFACE DESIGN: A HOLISTIC PROCESS

5.1 Roles

In defining this process, it first is advisable to determine the roles. Often on an e-learning project, several roles exist, including instructional designer and graphic artist, as two that affect the interface. In the process I have defined, the design lead, or "design architect," is responsible for the learner's experience. This role definition places one person in charge of the learner's experience, and avoids the pitfalls of a design-by-committee approach. Further, it defines the graphic artist's role as doing the artwork as directed by the architect, rather than implicitly making larger decisions about the learner's experience. It is also assumes the programming or authoring can be done by either the designer, or by someone who reports to the designer.

5.1 Summary of the Process

In this section I describe an e-learning design and development process which is geared to produce a sound interface. While these project development stages are not designed solely for interface purposes, they are key parts of a holistic approach to e-learning design which, in my view, is necessary to ensure a good interface. Several content-development stages from an overall process have been omitted, in the interests of focus. At each step in this process, I note the particular relevance to interface design.

Step 1: Specify the audience and its characteristics
- *Definition/Description*: At this step, the design architect should be as specific as possible about who the audience for the program is, including primary and secondary audiences.

- *Directory Assistance example:* The primary audience is newly-hired directory assistance operators; the secondary audience is underperforming experienced operators who are deemed by their supervisors to need refresher training. All operators have a high school degree but only some have college degrees. The operators are only somewhat experienced with computers, as a rule; the age range is wide, and most of the operators are female.

- *Relevance to Interface Design:* The characteristics and experience of the audience, taken broadly, often should impact the design of the interactions and also the interface. For example, an audience of preschool teachers may prefer colors and fonts that are more akin to what they see in the classroom; a mostly-male audience of warehouse workers might (and did, in one project) prefer darker colors and very direct, non-stylish language. Such stereotypes can, and should, play into interface design.

Step 2: Articulate the goals
- *Definition/Description:* Define the goals of the program clearly and concisely; this step covers what learners should be able to do at the completion of the course.

- *Directory Assistance example*: Learners will be able to look up listings quickly and accurately, averaging under twenty seconds per call with less than one wrong listing per fifty calls, will know what questions to ask the customer. [Note: customer service skills such as politeness and learning

what phrasing to use are not within the scope of this project, though should be reinforced as relevant].

- *Relevance to Interface Design:* The entire learning environment and learner's possible actions must be designed around the goals.

Step 3: Develop concept and learning methodology

- *Definition/Description:* In this step, the overall concept is established for the program, meaning what methods will be used and what the learner will do in the program.

- *Directory Assistance example:* The Directory Assistance Training Course will be a learn-by-doing simulation, in which the learner takes simulated calls in audio, and must look up the listings quickly and accurately. A tutoring component will interject when the learner makes mistakes, and will be available to guide the learner through the tasks.

- *Relevance to Interface Design:* As with Step 2, this is a high-level step which influences the entire program.

Step 4: Develop specific teaching points, scenario overviews (in the case of a training program)

- *Definition/Description:* In this composite step, the specific things the learner will be able to do after the training are articulated, along with (in the case of a learn-by-doing simulation) the scenarios the learner will be asked to encounter in order to sufficiently cover all teaching points. This step is dependent on the concept determined in step 3; for example, if we were evaluating a performance-support program, this step would not be the same.

- *Directory Assistance example:* Specific things to teach in this program include: how to look up a standard residential listing; how to efficiently look up a residential listing when the name has less than four letters; and how to play a recording of a phone number for a customer. Scenario overviews cover the different types of listings (in this program, a less-than-four-letter residential name would have one or more scenarios dedicated to it).

- *Relevance to Interface Design*: The teaching points and scenarios again help drive the overall design, including the interface. This is the last of the high-level steps before interface work is begun.

Step 5: Design sketch: Wireframes

- *Definition/Description:* The process here involves sketching out wireframes—black and white sketches showing the layout and positioning of items on the screen. The reason for wireframes rather than graphics at this step is to keep the focus on the learner's conceptual actions. The designer should focus on actions and layout, rather than graphics, at this stage, and design the primary (most heavily-used screens) first if possible.

- *Directory Assistance example:* The main screen, as shown in Figure 1, was sketched out first, and the few auxiliary screens later.

- *Relevance to Interface Design:* In this step, the specific interface design is finally begun.

Step 6: Sample screens/storyboard (full graphics)

- *Definition/Description:* A set of sample graphical screens are produced, showing the intended look of the screens the user will see. These are often best produced in a sequence that shows a part of a realistic interaction that a user might have. The design architect should look for several things at this stage (and this stage is often iterative), including the following:
  - The graphical look should support the tasks and layout from the wireframes, and in an ideal world improve upon them. This means that, for example, the relative sizing of things is as defined in the wireframes.

o The colors should be appropriate for the audience (and, in a client project, the client's needs), and color should further allow, for example, the learner's eye to be drawn to the parts of the screen the designer needed it to be drawn to.

This generally should be the first interface design stage that outside groups—the target audience and client—should be able to see and react to. Their feedback should then be incorporated; specific methods for doing so are beyond the scope of this paper, but the philosophy is that the feedback should be taken seriously, but not literally, and many questions should be asked. For example, if a reviewer said they thought a particular button should be blue, one would want to uncover why they thought so, and work the feedback into a holistic set of design changes to address the goals and concerns raised by the reviewers, rather than specific solutions raised. While not listed here as a separate step, changes will generally be made after the review, and either the screens will be reissued or the changes can be incorporated into Step 7, depending on the degree of change required.

- *Directory Assistance example:* The Directory Assistance sample screens showed the main screen, as in Figure 1, and then also showed examples of tutoring feedback, tutoring guidance, and the "Talk to the Customer" button set. It was crucial that the computer screen (the white/blue/green area that is exactly what a real operator would see) was a reasonable size and was the focus of the screen; the text on the buttons needed to be clear and readable but not too large, and the buttons needed to show dimension in order to make it clear that they were clickable.
- *Relevance to Interface Design:* Another step that is focused solely on the interface itself.

Step 7: Single-path demonstration
- *Definition/Description:* At this stage, an online demonstration is produced, showing a carefully-constructed path as the learner would interact with the program. The path should be designed to tell a realistic, compelling story of the learner's interaction, and should very specifically explain what the learner does at each stage of the demo and what happens on-screen, so that the graphics and programming, however they are done, are exactly what the designer had in mind.
  o The demonstration should be shown from the point of view of the learner, so the audience viewing the demo can get a feel for what it is like for a learner to use it.
  o The designer, when reviewing the demonstration, should look to be sure all screens are correct and that the interactions seem smooth to a learner; for example, the screen does not "jump" (no graphics move slightly, accidentally, on-screen), and unnecessary portions of the screen are not "redrawn."
  o It is not necessary for all features to work correctly technically at this stage—it is only necessary for things to appear as if they are working to the viewer. For example, a PowerPoint slide show often can make a good demo, though this demo would only function by following a single path no matter what action a learner tried. For demo purposes, this is sufficient and this method is often a good use of resources.

The demonstration is another major deliverable that the target audience should be able to see and react to. Feedback should be treated as in Step 6—seriously but not literally, with an emphasis on goals.

- *Directory Assistance example:* The Directory Assistance single-path demo showed a caller asking for a residential listing, the phone number for Dennis Thompson in Deerfield, Illinois. The learner in this demo path asked the tutor Now What? and How Do I Do That? in order to show off how the tutor could be helpful, made two errors in order to show off the feedback component, released the listing to the customer successfully, and then checked performance.

- *Relevance to Interface Design:* Here we see the interface in action for the first time.

Step 8: Detailed design document
- *Definition/Description:* This document should describe the program's behavior completely, including all screens and actions (building on the work done in previous steps). All graphics should be approved and referenced in this document as well (so a smaller step, omitted from our major steps but a prerequisite to this one, is to sketch and then receive and approve all artwork from the graphic artist). The design document is used by the programmer to implement the program.

- *Directory Assistance example:* The Directory Assistance design document included all screens and the rules for all actions—explaining, and showing as necessary, exactly what should happen after each learner action.

- *Relevance to Interface Design:* This document describes the entire program, including the screens, in detail.

Step 9: Prototype version and primary usability testing

- *Definition/Description:* The prototype is a working program, generally with most of the features implemented and most of the content included. The prototype is particularly crucial for everyone, beginning with the design architect, to first get to interact with the real program. This version, after changes from the architect's review, generally should go to the target audience for the most significant usability test, to determine how clear it is to use.

- *Directory Assistance example:* The Directory Assistance prototype included only ten of the fifty scenarios (simulated callers), and all features except for the peripheral ones. The reason for the small number of scenarios was that the scenario structure was repeated frequently, so it was more efficient and equally useful to build out ten rather than more.

- *Relevance to Interface Design:* We see the full program, with a few exceptions, in working condition, and can test the usability and success of the program and interface.

Step 10: Implementation completion and pilot testing

- *Definition/Description:* The program is completed and ready for final testing. Interface issues may still arise, though, because there is new content, which may not be clear. Sometimes it is arguable whether to categorize such points of learner confusion under "interface," but under our holistic process, any concerns certainly need to be addressed.

- *Directory Assistance example:* The Directory Assistance program included all fifty sample calls when it was rolled out for pilot testing.

- *Relevance to Interface Design:* Here we see the full program in its entirety, for final testing with a small group, gathering one last round of interface feedback.

After pilot testing and the incorporation of any final feedback (re-testing if necessary), the program is completed and ready to be released.

6. CONCLUSION AND FUTURE DIRECTIONS

The process I have outlined has been successful across over a hundred projects. There seem to be several reasons for its success:

- The interface is designed in the context of the learner's tasks, for the purpose of supporting tasks to be done by the learner;

- The interface is designed by a design architect with experience in online learning design and with a deep understanding of the tasks he is asking the learners to perform;

- The high-level goals of the project, and the characteristics of the audience, are considered relevant all throughout the design process.

I suggest that e-learning interface design, and e-learning design as a whole, would benefit from adopting and refining the process above. This process can be fleshed out in more detail and can be further customized for different types of e-learning; for example, the specifics of a process for a learn-by-doing simulation would differ in place from those for an information reference. It is my hope that this process and others can provide a basis for the future development of more-effective e-learning.

REFERENCES

Guralnick, D., 2000, "A Step Beyond Authoring: Process-Support Tools." *WebNet 2000, Association for the Advancement of Computing in Education*. San Antonio, TX, 468-471.

Guralnick, D., 2006, "How to Design Effective, Motivating User Interfaces." *American Society for Training & Development TechKnowledge Conference*. Denver, CO, 124-142.

Nielsen, J. ,1993, *Usability Engineering.* Academic Press, San Diego, CA.

Nielsen, J., and Molich, R., 1990, "Heuristic Evaluation of User Interfaces." *Proceedings of the ACM CHI'90 Conference.* Seattle, WA, 249-256.