

Interface Design for Educational Multimedia

One of the most exciting parts of the software development process is designing the part of the software that users will see and interact with. The interface is the visible personality of the software. It bridges the gap between the programming that makes the software work and the human using the software.

This guide describes the software development process, including the interface design phase. Interface design is not an isolated task. It is only a part of the process of product design, and every phase of that process affects every other phase. This guide places interface design in the context of software development.

Principles of Interface Design

The point of an interface is to allow the user to interact with the software without having to be reminded that he or she is working with a machine. Everything you do as interface designer, therefore, should contribute to making the users' lives as pleasant and easy as possible, at least while they are using your software.

The *principles of interface design* are the guidelines that should be kept in mind while designing any user interface. There will be times when you, the designer, will stretch these guidelines, but for the most part, especially if you are new to interface design, you should follow them.

Simplify the users' tasks.

One of the first steps you will take as you design your product is to define what the users will want or need to do with your software. Every interface-related decision you make should make it easier for the users to accomplish their tasks. If they will need a certain set of buttons frequently, make sure those buttons are on each screen and are always in the same place. If users will need to move between screens often, make sure your navigation system supports such movement.

Be consistent.

This is one of the hardest parts of designing a complex interface, because as designers we can only do one thing at a time. By the time you've designed the last screen, it is possible that small changes will have crept into your design - changes that you may not notice, especially if you have been working on your design for days, weeks, or longer. However, the user will notice and be confused if your buttons suddenly change shape, size or color, even slightly. Likewise, confusion will result if something that is clickable in one place is not clickable in another place. Be careful to stick to your theme and to the layout system you design. Make sure that elements of your interface which function alike, look alike, and that elements which don't function alike, don't resemble each other too closely. Build on the expectations that users will have from using similar software: we expect menus in the top, left-justified. We expect large white areas with scroll bars to be editable or at least readable text fields. If we see an I-beam cursor, we expect to be able to edit text. If we see a pointing hand cursor, we expect that clicking will do something. A spinning ball or ticking watch will assure us that we should wait, and something will happen soon.

Give feedback.

Visual feedback is very important to users as they are working with software. When someone moves to another screen, for instance, visual feedback that they have moved is very helpful. Indicate the new position with a highlighted button or a new title - something visible besides simply changing the text in the body of the screen. Buttons that highlight when the mouse rolls over them are appealing and effective. If you are incorporating sound in your software, you

can use auditory feedback as well: a *click* when a button is pressed, a *whoosh* when a detail view is accessed.

Be forgiving.

If a user makes a mistake, your software should compensate and the mistake should be easy to remedy. If, for instance, the Erase button is right next to the Save button (probably not the best place to put it anyway), a message asking the user to confirm the command is helpful in case he or she has mis-clicked. Similarly, users should be given the option to skip or interrupt long animations, narrations or video segments. Avoid dead-ends in your software, where the user must retrace his or her steps.

Never trap the user in a situation where he or she must answer a question correctly to continue. If an incorrect answer is selected, present the correct answer and allow the user to move on.

Recognize the user's limitations.

If there is a task that your software can do for the user, such as performing calculations, build it into the software instead of requiring the user to do it off-line. The user should never need to remember long lists of choices or steps. If several steps are necessary to complete a task, give the user an overview, and then repeat each step as it is appropriate.

Make sure that the purpose of buttons and icons is clear. It is very frustrating to have to click a button in order to find out what it does, especially if the result is an unwanted action.

Group information appropriately. It is difficult for users to remember long lists of menu items, so group similar items together and give each menu a descriptive name. This will minimize the amount of time a user must spend searching for a particular menu item.

Be flexible.

Provide shortcuts in places where experienced users might want to use them. Provide various ways of learning or doing something so that different users will feel comfortable.

For example, most word processors provide several ways of saving a file: keyboard shortcuts, menu items, and buttons on a toolbar. Since these are different ways of accomplishing a common task, users may select the one that suits them best.

Remove obstacles and redundancy.

Remove frustrating obstacles, especially to frequently-used tasks. If a confirming dialog box pops up each time the user tries to save, he or she will quickly get frustrated. Remember that clicking a mouse button is a strain on the wrists, and try to reduce the click count for common tasks.

Avoid redundant controls. If you offer the user multiple ways to access your help screen on every screen, he or she may become confused about which is the right one to use. The screen space would be better used another way. In the example above, the three ways to save a file are not confusing. Two buttons to save, located in different places, would be. And consider how often users are uncertain whether to use *Save*, *Save As*, or *Save a Copy* from the list of choices in the File menu.

Be explicit.

If something is clickable, make it look clickable. If users will need certain information or skills to accomplish a task, make that plain before they get involved in the task. Don't make users guess what the elements of your interface are for. Your interface should help them get their tasks done rather than getting in their way.

*A road that does not lead to other roads
always has to be retraced, unless the
traveler chooses to rust at the end of it.*

Tehyi Hsieh

Chinese Epigrams Inside Out and Proverbs

Consider culture.

If your software will be used by a wide audience, be very careful about using symbols or metaphors that are strongly rooted in culture. Instead, use symbols which are recognizable internationally. Be careful; it is easy to unintentionally use offensive language or images.

Remember the user.

Everything that you do, everything you design, should be based on the needs of the users. You are not designing for yourself – you are designing for the people who will sit down in front of your software and try to use it. Remember that they are there to do certain tasks, and if you make it difficult, they will not use your software.

The Aesthetics of the Interface

The ideal interface is easy to use and aesthetically pleasing. The principles of interface design address issue of ease of use. The aesthetics of the interface address the issue of designing something that is pleasing as well as functional.

Clarity

There is usually more to put into a software screen than will fit, so select your elements with care and make their relationship to each other very clear. Use spatial grouping, color, and contrast to indicate which elements are related. When you perform user tests, take notes about where the testers make mistakes or form mistaken assumptions, and clarify those parts of your interface to eliminate misunderstanding.

Sequence

The eye travels over the screen from the upper left to the lower right. Keep this in mind when placing interface elements. The most important elements should be closer to the top and closer to the left. If steps must be followed in a certain order, the first step should be uppermost and leftmost, with subsequent steps following.

An example of poor sequencing can be seen in many forms or surveys on the web. Often, after filling out the survey, the user automatically clicks on the button in the lower right, which is where the last step is expected to be. However, in many forms, the *reset* button is located there, and the *submit* button is located to its left. By the time the user has figured this out, the data has been erased from the form.

Proportion

The largest items on a page or screen should be the most important, since that is the perception users will have. Buttons which are often-used or which need to be accessed quickly should be large as well. The access time for a small button (the time it takes the user to find it, move the mouse to it, and click it) is much longer than the access time for a larger button. Help users to shorten task time when you can.

Balance

Finally, all the elements that you have to fit into your design clearly, sequentially, and in correct proportion should work together to create a balanced screen. Imbalance creates feelings of uncertainty and unease, which can adversely affect users' ability to perform the tasks they are using your software to perform.

Creating balanced layouts takes practice. Look at as many interfaces as you can. Look at web sites, shrink-wrapped software, even video games to get a feel for what is being done. Try using a grid system in your own layouts to help you balance your screen elements.

15. How much time did this take?

- 10 minutes
- 20 minutes
- 30 minutes
- 40 minutes

Thank you for filling out our form!



Most web forms do not need a reset button. If the user does not wish to submit the form, he or she may simply close the window or go to another page.

Designing the Project

What is the product you will be creating? Who is your primary audience? What are the goals for this product – what will the users need or want to do? What authoring tool(s) will you use? The answers to these questions should form a project specification which you can use to guide each step of the development process. Although it seems overwhelming at first, if you answer each question in turn, you will have a strong base on which to build your project.

Defining the Problem

Define the product.

What is the problem you are addressing? How will your product help solve that problem? Write down in clear terms the answers to these questions. Think about the primary purpose of the product. Is it promotional? Instructive? It is often difficult to pinpoint what the product will be before it is begun, but your tasks will be easier if you write down your expectations as you begin.

Think about how you plan to deliver the product. Will it be a web site? A Java applet that is downloaded from a web site? A standalone application delivered on CD-ROM? Write down the delivery medium (web, CD, disk...) and the type of product (web site, applet, application, presentation...).

Define the audience.

This is one of the most important steps of interface design. Write down, as clearly as you can, who your audience is. Write down your assumptions about their skills. Then go find a few people who are typical of your audience and test your assumptions to make sure you're on track. Everything you design must be tailored to the people who make up your audience. Do surveys, test groups, whatever you can afford.

Remember that your audience are people. During software design, these people are almost always called users. Always keep them in mind and remember that they are usually not computer experts, that they do not want to puzzle out how to use your product, and that they are easily frustrated.

If your audience is widely varied, you will find that they have mixed skills. In such a case, your task is harder because you need to create software that is easy to use, even by inexperienced people, without being boring, insulting or tedious for the more experienced.

Define the goals for the product.

Defining the goals can be the most difficult part of product design. As you begin writing goals, consider the content that should be or must be in your product. To help clarify your goals, think about the people who will be using your product (your audience). Write down what you expect them to need to do or want to do, and what qualities the product should have to help them succeed. How does the content of your product relate to those tasks?

Sometimes there are other requirements imposed on your product. Perhaps the content author(s) have certain restrictions for how their content may be used. Make note of any special requirements that are present in your case.

The goals for your product should reflect the outcomes you expect the users to achieve. For instance, if your product is a piece of promotional software describing your department and its services, one of the goals could be as follows:

Our audience will know how to access our such-and-such service.

Focus on the audience rather than the product as you write the goals. For instance, the above goal could also be written as follows:

Our product will demonstrate how to access such-and-such a service.

Writing the goal this way causes you to focus on the product for its own sake, instead of on the needs of your audience. Take every opportunity to remind

yourself that you are developing for your audience, and you will create a better, more usable product.

Choose the authoring tool(s).

An authoring tool is software which you use to develop your product. If you are creating a web site, your authoring tools might include a web page editor such as Claris Home Page, a text editor such as WordPerfect, and an image editor like Adobe Photoshop. If you are creating a multimedia demonstration for CD-ROM, you might use Hypercard, Authorware, or Director, along with software for editing text, images, sound and video.

Sometimes your choice of authoring tool is limited. Limiting factors may include the medium by which your product will be delivered, the nature of the product itself, and the tools you happen to have or be able to acquire. Since any tools you select will impose restrictions and create possibilities for your design, add a list of your authoring tools to your product specification.

Develop the flowchart.

After you have answered the questions of what your product will be, who your audience are, what tasks they will want or need to do, and what tools you will use for development, you are ready to begin flowcharting your project.

The flowchart is a map of the product's structure, of the paths that users can take as they navigate it. The flowchart's primary purpose is to help you develop and chart your product's usability. Flowcharting forces you to think about the product's structure. When you start with a clear picture of the product's structure, your audience will find it better organized and easier to use as a result.

To create your flowchart, begin with the first page or screen users will see, and try to visualize each choice they will have. Map out the screens these choices lead to. Keep mapping until you have charted the entire scope of the product as you believe it to be.

The flowchart should just indicate the content of each screen and the path or paths through that content. Keep it simple – don't put in too much detail at this stage. Just make notations of the pages or screens, and of how to move between them. Your initial flowchart may not bear a lot of resemblance to the finished product, but it is an important document which will guide you in the development stages.

Interaction Design

How should the product work? This is the main question which is answered in the interaction design stage. Your work during this stage is to develop a guidance system, design the navigation, and define every screen that will appear in your product. You will also create a storyboard at this stage.

Designing the Guidance System: Orienting Your Users

The first few screens or pages should indicate clearly to the user what is in the product and where to find it. What are the options the user will have on the first screen? On each of the screens one link away from that screen? You will specify choices that appear on later screens when you create the storyboard. The choices you make now for the first few screens will set up the system that users will learn and expect to follow while using this product.

Designing Navigation

Using your flowchart, identify points at which the user controls the path. List the different types of controls needed in the product (controls to move between main topics or sections, within topics or sections, controls that take the user to help or information or map screens, controls for printing, editing or saving, if any). Create a consistent system for navigating the product based on each type of screen.

Designing Each Screen

Where will elements be placed? Which screens have similar elements? What do the controls look like (in general)? Make a list of each screen or page, including in your list the elements present on the page and a rough idea of their location. Sketch any special details or important features.

Developing a Storyboard

Using the list of screens, create your storyboard. Draw or sketch each screen and block out its contents in a rough way. Make notes on the storyboard about sounds, video clips, or other media that will be present on each screen. Specify when such media appear (automatically, or at the touch of a button?), how long they last, how they can be stopped early, and where on the screen they will show.

Notes about navigation and user-controlled choices should also be included on the storyboard. What choices will be present on each screen? Where will the controls be located? Keep in mind that controls that look the same should serve the same function and should usually be located in the same place from one screen to another.

Check Your Goals

As you progress through the interaction design stage, be sure to check every decision you make against your goals. Will each decision will move the product closer to or farther from achieving the goals you have set for it? If a decision is not in accordance with the product's goals, do not be afraid to change it. It is much easier to make changes during development than after release.

Interface Design

Designing the interface, or screens, for your product is challenging and exciting. This is the part of the product that will be visible to the users, the part that they will relate to and interact with. It is the face of the product.

You will find that interface design is not a linear process. The several tasks that make up the interface design process tend to all happen at once, and each has an impact on the others.

Define the visual theme and style.

Select and define a theme that is consistent with your goals, your users' expectations, and your content. For example, if your software is an instructive piece about dinosaurs aimed at six-year-olds, your theme and style should be playful, inviting, and lush. You may choose to use dinosaur-shaped buttons and prehistoric images as backgrounds.

Think of the theme as the personality of your product. Try to develop a personality that is appropriate for your audience and your content. Themes are often derived from a particular topic or context; look to geographical locations, nature, technology, or certain time periods for ideas and inspiration.

The style you select will help unify the elements in your designs. If you have existing graphics, take your style from those. If you have a collection of graphics of different styles, try to select or develop one based on similarities you find in the images. The designer(s) who are working on the product will be strongly influential in selecting and/or developing its style.

Design system of screen layouts.

If you are developing a product that will stand alone, your main concern will be that the screen layouts work together within the product while clearly differentiating the sections. If your product is one of a family of similar products, you must also take into account the family resemblance among the products. If this is the first of the family, take care to consider how its look and feel will translate to the other products in the group. If it is not the first, you will have to derive its look and feel from the other product(s) that already exist.

The Law of Digital Media:

- *High-quality images & sound*
- *Able to run on most computers*
 - *Acceptable performance*

- pick any two -

Ray Kristof & Amy Satran

Interactivity by Design

There are conventions of layout that you should be aware of. Information is easier to absorb and retain if it is chunked (divided into discrete sections) rather than presented in one big clump. Remember that English is read from top to bottom and from left to right. Your screen layouts should reflect this, especially where the sequence of steps or of information chunks is important. If you use “continue” or “more” signals, these should be placed at the lower right, where the eye will naturally rest after reading or viewing a screen.

Items which are larger in size will be perceived as more significant than smaller items; be aware of this perception and use it to your advantage. Likewise, things that are on top are perceived as more important than things that are on the bottom. Be careful that you do not accidentally convey incorrect messages about the importance of various controls, images, and textual elements.

Remember these three things as you develop your designs: keep it simple, keep it consistent, and know when to break the rules.

Create structural elements of each screen.

Backgrounds, fields and windows are basic elements which delineate the structure of a page or screen. You can use them to set up expectations of where to find controls and media elements in your software, even before such elements are placed.

The backgrounds you select should enhance without distracting. If they are too noticeable, the flow of your layout may be lost. Select pictures that will not interfere with other elements on the screen such as text and images. Create low-contrast backgrounds that provide visual interest without calling attention to themselves.

As you design text windows and control panels, think about where they will go and what will be behind them. If buttons are placed directly on the background, they may appear to float. Sometimes it may be difficult to make out their edges and they may fade into the background. Placing a block of semitransparent color behind them can alleviate these problems.

Similar problems can occur with text. If the background image is detailed, it will probably interfere with the text, making it difficult to read. Placing a semitransparent white block behind the text area can help make the text stand out. If you are consistent with design issues like these, the blocks of color behind controls and text will create expectations in the user’s mind about the function and behavior of similar elements, making your software easier to use.

There are conventions about placement of some elements, also. Controls are usually located around the edges of a screen, and menus are usually at the upper left. As you plan for placement of items, think about registration, or their position from one screen to another. It is very distracting when buttons and text fields jump around as viewers move between screens. Make sure the placement of identical items is perfectly aligned from screen to screen. You may find it helpful to use a grid layout as you place elements.

Create control elements.

Refer to your flowchart and create the controls that users will need to move between screens and to operate any functions in the software. As you design controls, remember that their function must be obvious at first glance. If you cannot design an icon that will be obvious to your audience, add text to the control to clearly indicate its function. Controls should be located for easy access, especially if they are frequently used. Their size should be proportional to their importance.

Create media elements.

Create or gather the images, sounds, text, video, and any other media elements you plan to include in your software. Elements in hardcopy (printed form) will

need to be converted to digital form by scanning or typing, or recreated completely in digital format. Save copies of everything on backup disks.

If you are scanning images, scan them in at high resolution, even if you know you will be using them at screen resolution. It is far easier to edit a high-res scan, which has more information than you need, than to edit a low-res scan which may not contain enough information. Keep track of any changes you make to your originals so that you can recreate the process if you need to with the same or a similar image. Keep copies of all your original scans for later revisions or changes.

Keep in mind the restrictions you noted during the product design phase as you create your media elements. If you have restricted the number of colors to 216, for instance, make sure that the color palette of every image you use has been reduced appropriately.

Create prototype screens.

Before programming or authoring the product, create a paper prototype and run it through some user tests to make sure your designs are as good as you can make them. To make a paper prototype, simply draw each screen on a sheet of paper, and draw each dialog box on a post-it note. Draw screen backgrounds on white paper and draw individual screens on tracing paper.

See the section entitled *Rough Usability Testing*, below, for more information on performing a usability test with a paper prototype. At this point you should also perform an in-depth review of your goals for the product, checking each design and control choice against your list of goals. Revise any choices that do not move the product toward achieving its goals.

After your design assumptions have been borne out, or after you have made appropriate changes based on the results of the usability test, create graphical prototype screens. Create each screen with image-editing software, placing backgrounds, controls, and media elements as appropriate. Use these designs in a group critique.

The Critique

The critique is one of the most emotional and most difficult phases of the development process, but it can also be one of the most rewarding. There are two kinds of critique: the personal critique, and the professional critique. Both require patience and practice in order to be beneficial.

Personal Critique

The personal critique is an ongoing process that happens between the designer and his or her work. It happens when the designer arrives in the morning and sees the results of yesterday's efforts. It happens after lunch, when the designer gets a fresh look at the morning's work. Any time the designer returns to the work after thinking about something else, a personal critique may take place.

In order for the personal critique to be useful, the designer must learn when to step away from the work. Only by looking at it with fresh eyes will he or she be able to take an objective view. To perform a personal critique, leave the work for at least half an hour, longer if possible. When you return to the work, look at each screen or element you have completed. Ask yourself these questions, and try to answer them as honestly as you can:

Does this screen/element fit the chosen theme of the product?

Does this screen/element fit the selected style of the product?

Is it appropriate for the intended audience?

For each control: is its function clear and easily apparent?

For each layout: do the parts fit together in a balanced way?

It is not unusual to find that some designs will need to be revised after a personal critique.

Professional Critique

The professional or group critiquing process requires an atmosphere of humor and trust. It is best to have the critique in a comfortable, creative environment, preferably not the designer's office. Neutral territory is best. Invite the other members of the product team: the content experts, other interface designers and graphic designers, programmers, and the project manager(s). This process is about improving the product, and anyone who has a stake in making the best possible product should be there. Project the screen drawings if you can. If not, use color printouts, or even have everyone gather around a computer screen while you step through the designs.

Set up expectations, especially if your team is unused to the process of a professional critique. You may wish to explain the process of giving and receiving critique (see below) to help the session go smoothly. Explain that first you will show all the screens, describing what each one does (if there are people in the room who do not know), and that then you will go through them a second time, when others should give feedback and suggestions. If there is a particular area you want help with, mention it and ask people to pay special attention to that problem.

Host professional critiques when you are blocked on a problem, when you have made major progress or changes, and when the graphic prototype is nearly complete. Not every professional critique needs to involve the entire team, although at least one should. If you are blocked on a particular problem, you may wish to meet only with the other designers and/or the programmers, or the content experts. Let the situation be your guide.

Giving Critique

The goal of those giving critique is to offer good advice in a way which will allow it to be heard. While you are giving critique, remember that you are talking about the product, not about the designer. Remember also that the designer is very close to the work and that his or her emotions are probably bound up in it fairly strongly. However, the designer is hosting the critique in order to improve the work, so it is important that you be as honest and as helpful as you can.

One strategy for giving critique is to try to make two positive, supportive comments before offering a criticism or suggestion. This will only work if you are sincere; it is very easy to tell when the positive comments are shallow and only perfunctory. If you cannot think of anything positive to say – which hopefully is not the case, or you may be working with the wrong designer – try to phrase your suggestions in a positive way. Build on what is already there in the work, rather than dismissing it entirely.

Sometimes you will look at a design and see what you know from experience to be an error or a flaw. If this is the case, be clear in your feedback that you are not making a best guess. Specify that you have experience with this situation and give reasons for your suggestion. If, on the other hand, you have a feeling that something may be wrong but you aren't sure why, specify that you are making a best guess and that you are not speaking from direct experience. This will help the designer weight your suggestions and decide what to change as a result.

Be humorous but not at the expense of the design or designer. Lightheartedness is valuable during a critique, but indiscriminate teasing is not. Be open: if you make a suggestion and someone modifies or improves it, listen to the new idea and make an honest judgement about it. The critique is not about how skilled or how clever you are; it is about improving the product.

Focus on the quality of the product as much as possible. This can be difficult if there are severe economic or time pressures involved. Try to see the work as a whole; if the two main colors are not your favorites, try to put that aside and

think about whether they work with the theme and style of the product.

Finally, if you have nothing to say, say so. You don't have to offer suggestions. You may be totally satisfied with the design, or you may not have anything to say for whatever reason. Be helpful if you can, but don't feel you have to come up with something to say. Remember that this is a professional session and act appropriately.

Receiving Critique

Receiving critique is not easy, especially at first. If your team is inexperienced in giving critique, it will be even harder. However, it does get easier with time, especially if your team works together often and is willing to develop critiquing skills.

When you host a critique, recognize that you have taken your designs as far as you can on your own, and that you are seeking fresh eyes and fresh ideas to help take it even farther. You are there to listen to what your team has to say, not to defend your design decisions. Suggestions that you hear are not attacks on you or your skills.

Remember that the critique is not an exam to which you know the answers and your team does not. If you approach the critique expecting someone to say something in particular, whether good or bad, you may not listen to what they actually do say. Keep an open mind and really listen. Take notes. If you are blocked on a particular problem, an answer may come to you during critique. When you see others looking at your designs, it often helps you to look at them in a more objective way.

Finally, remember that the critique is not about you, or your skills, or your self-esteem. It is about making a better product. If you are shy or lack self-confidence, your first few critiques may be painful; but after a while, they will become easier. Be professional and gracious, and your work will benefit.

Rough Usability Testing

Usability testing is a way to allow the developers of a product to see how someone who has no experience with the product wants it to work. By the time your product is ready to usability test, even paper testing, you are far too close to it to look at it objectively. You know what the controls do. You know why it looks the way it does. You designed the functionality, so you are easily able to accomplish any task that you imagine a user would need to do.

However, someone who has never seen your product before does not know all the things you know about it. If you watch people using a prototype of your product, you can learn an amazing amount about how it will do once it is released. If there are major flaws, you will be able to correct them. If you have made assumptions about your audience and how they will perceive your product, this is your chance to test those assumptions.

Paper Testing

To perform a user test with your paper prototype, find a group of testers who match your audience profile as closely as possible. Conduct each test individually. Place the first drawing in front of a tester and give a brief explanation of the purpose of the software. Explain that you are testing your design, not the person's ability, and that you want the tester to think out loud as she or he works through each task. Instruct the tester to "push" the buttons with a finger so that you can take note of the actions he or she takes. Stress that the tester is free to stop at any time if he or she becomes uncomfortable or bored or just wants to leave.

Give the tester a predetermined task to accomplish. Give each tester the same series of tasks, so that you can compare results. Explain to the tester that you will not offer guidance or help, and that you will only observe as he or she

works through each task. Stress that there are no wrong actions. Everything the tester does will be useful to you.

Then watch as the tester attempts to complete the assigned task. When he or she presses a button, switch drawings or insert a prepared sticky note to indicate what would have happened in the software at that point. Write down the path the tester takes, each action he or she makes, and any helpful thoughts that he or she voices. If the tester gets completely stuck, you may move on to the next task or explain what action could be taken next, as the tester prefers.

Allow the tester to work through each task at his or her own pace. If the tester wants to stop early, don't force him or her to continue. If the tester gets stuck and isn't verbalizing his or her thoughts, ask leading questions, like "What are you looking for? What did you expect to see or do?"

When the tasks are complete or the tester decides to stop, thank him or her and ask if there are any questions he or she would like to ask. It may be that some testers want to know why something has been designed a certain way, or what they ought to have done in a certain situation. Talk it over as much as the tester would like.

Write down any other notes while they are fresh in your mind, and before you begin with the next tester.

Testing with Screen Drawings

You can perform similar tests with printouts or on-screen versions of the product that you have created in an image-editing program. For screen tests, allow the tester to click buttons with a mouse. You may be able to use imagemaps or similar low-overhead technology to create a rough mockup of the expected functionality. The same guidelines for working with testers apply with screen drawings as with paper prototypes.

Testing the Working Prototype

If possible, have your working prototype tested by professional software quality engineers. If that is not possible, round up your testers again (you may wish to mix new testers with ones you have used before) and allow them to work through another set of tasks while you watch. You may find it helpful to videotape the sessions if the testers will consent to being taped.

Using the Results

Be sure to use the results of your test sessions. If testers consistently had trouble with a particular part of your prototype, change it. If a button was unclear to several testers, use the information from the test to redesign it to make it more clear.

Sometimes designs that are very dear to the designer simply do not work for the users. When this is the case, don't be afraid to change the design. Keep a copy in case you can use it later in another way, but listen to your testers and act on the information they give you.

Example: Successful Design

HemoglobinLab

The product to be created was an interactive, instructive Java applet which would allow college-age biology majors to study mutations in human hemoglobin. The primary audience is college-age biology majors. The secondary audiences are other college-age biology students and college biology faculty.

The goals of this product are as follows:

- Students will be able to perform virtual examinations on virtual patients using a variety of tools and methods;
- Students will be able to examine the protein strands that result from various mutations of human hemoglobin;
- Students will be able to edit and translate RNA strands to attempt to recreate the mutated strands they have studied.

This product was developed under a number of constraints, namely:

- It must be deliverable over the web;
- It must not require any special software other than a Java-enabled web browser;
- It must use only the 216 web-safe colors;
- It must be interactive, rather than tutorial-based;
- It must be flexible so that instructors can develop a variety of assignments around it;
- It must bear a family resemblance to several similar labs, already developed.

The authoring tools that were selected for this product include Illustrator, Photoshop, CodeWarrior, and associated graphics and coding software.

The attached flowcharts show the functionality of the applet.

The attached storyboard shows thumbnail sketches for each screen.

The development path of the screen drawings is available at the following address:

<http://www.pixeldust.com/rachel/hblab/>

Example: Not-So-Successful Design

HemoglobinLab Web Site

The product to be created was a web site to allow access to HemoglobinLab and its supporting textual material. The primary audience is college-age biology majors. The secondary audiences are other college-age biology students and college biology faculty. The goals of this product are as follows:

- Students will be able to access HemoglobinLab easily;
- Students will be able to access related materials, including assignments, help on using the applet, and background information easily.

This product was developed under a number of constraints, namely:

- It must be deliverable over the web;
- It must not require any special software other than a web browser;
- It must use only the 216 web-safe colors;
- It must be easily navigable;
- It must be clearly identifiable as being related to HemoglobinLab;
- It must bear a family resemblance to several similar web sites, already developed.

The authoring tools that were selected for this product include Photoshop, BBEdit, Javascript, and associated graphics and editing software.

The attached flowcharts show the functionality of the web site and its relationship to the main Biology Labs On-Line web site.

The attached storyboard shows thumbnail sketches for the screen layout and two sample screens.

The Problem

The problem to be addressed is the navigation system for the web site. At present the site is navigated by means of a frame on the left side of the screen, which is filled with links. There are two types of links that must appear in this frame: links that lead to points within the page displaying in the main window, and links that lead to other pages on the site.

Concerns about the existing system include the following:

- Frames are usually best avoided;
- When there are many links within the page, the links to other pages are lost at the bottom of the screen;
- The formatting of the links is confusing and not very attractive.

Your Turn

Here is an exercise in interface design. The product has been partly defined, and a list of constraints, a flowchart, and a list of existing media for the product have been provided for you. Your tasks are to design the product's interaction and begin designing its interface. We will host a critique and usability test for some of the designs as time permits.

The Problem

The product to be created is an interactive, instructive module which will help college-age students study a foreign language. The primary audience is college-age, first year language students. The secondary audiences are other college-age language students and college language faculty. The goals of this product are as follows:

- Students will be able to practice reading, writing, listening and speaking a foreign language;
- Students will be able to observe young people engaging in everyday activities in areas where the language is spoken natively;
- Students will be able to look up words and phrases and add new words and phrases to the dictionary.

The constraints for this product are as follows:

- It must be deliverable on DVD;
- It must run on both Macintosh and Windows platforms;
- It must use only 256 colors;
- It must be interactive, rather than tutorial-based;
- It must be flexible so that instructors can develop a variety of assignments around it.

You may select any appropriate authoring tools for this product.

An attached flowchart shows the functionality of this product.

Existing media include the following:

- 240 minutes of digitized video footage of young people interacting using the language;
- 180 minutes of digitized audio clips of words and phrases being spoken;
- A digital dictionary of this language to English;
- 60 still photographs of scenes from areas where this language is spoken natively;
- Ten short stories of varying difficulty written in this language.

On your development team are an interface designer, a graphic designer, a programmer/developer, three content experts, and a project manager. (Your group may assume any of these roles or add new ones if you wish.)

Interaction Design

Your tasks are as follows:

- Design a guidance system;
 - Design the navigation;
 - Design each type of screen;
 - Develop a storyboard.
-

Interface Design

Your tasks are as follows:

- Define a visual theme and style;
- Design a system of screen layouts;
- Sketch several prototype screens.

Glossary

audience: the people at whom your product is targeted.

authoring tool: software which is used to develop your product.

interface: the visible part of your product, with which people interact.

product: the software you are creating.

specification: a document about the product which contains the goals, constraints, design decisions, flowcharts, storyboards, and other information pertaining to the creation of the product.

usability: the functionality of a product; the way it works.

users: the people who will need or want to use your product.

References

Kristof, Ray and Amy Satran. (1995) *Interactivity by Design: Creating & Communicating with New Media*. Mountain View, CA: Adobe Press.

Nielsen, Jakob. (1998) *Introduction to Web Design*. CHI 98 Conference Tutorial Proceedings.

Watzman, Suzanne. (1998) *The Art of the Interface: Visual Ideas, Principles and Inspiration for Interface Designers*. CHI 98 Conference Tutorial Proceedings.

Woolsey, Kristina Hooper, Scott Kim and Gayle Curtis. (1996) *VizAbility*. Boston: PWS Publishing Company, an International Thomson Publishing Company.