



7

Designing the Information Architecture

A COOK IS SUPPOSED to have said: “The flour itself does not taste good, nor does the baking powder, nor the shortening, nor the other ingredients. However, when I mix them all together and put them in the oven, they come out just right for biscuits.” This chapter is about determining the right combination of information elements that will maximize the usability of the software system. The activity is called integration, and the resulting design is called information architecture.

7.1 What is information architecture?

The information architecture is the result of the integrated approach to information design. It is the blueprint for maximizing software usability via the integrated design of labels, messages, online support elements,

and printed support elements (see Figure 7.1). The information architecture identifies all the information elements users need and expect, describing each in terms of content, media, and form—all based on the needs and expectations of users.

The information architecture helps cut time and cost by ensuring that big design errors are found and corrected early. You know it is much easier to cut four chapters from an information architecture than to add three pages to a final draft. Just as the architecture plan for a building project coordinates the activities of many workers, the information architecture coordinates the activities of a number of writers. For the technical writing team, it makes information development a more professional endeavor.

7.2 Integration: what is it, why is it important?

For real usability, the entire user interface, including all aspects of every interaction with the user, should be designed as a single, integrated element of the system [1].

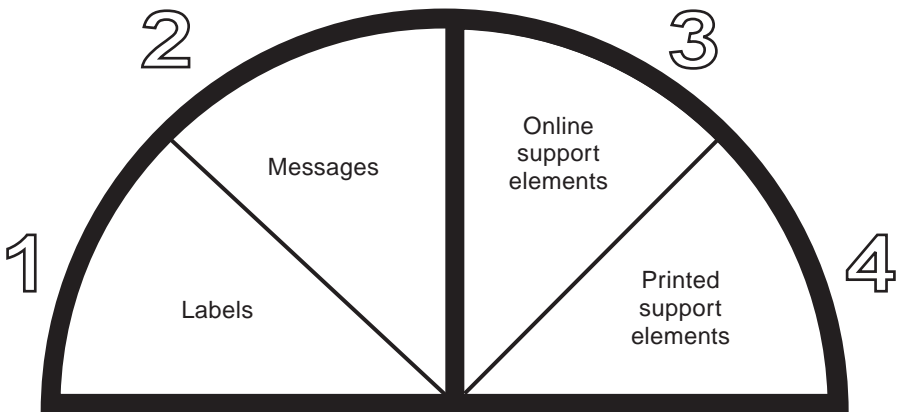


Figure 7.1 Four-tier information architecture.

If all the information elements you produce are good “documents” individually, you will maximize software usability, right? While talking about the reasons for hard-to-use systems, Jeffrey Rubin [2] says, “. . . this (non-integrated) approach is futile, since it matters little if each component is usable within itself. Only if the components work well together will the product be viewed as usable and meeting the user’s needs.”

Take for example a two-color, nicely produced user manual that has *all* the information about your software system. Though it also has that small piece of detail that a user is looking for, retrieving that detail will take a longer time because the manual is bulky, probably with many pages of unnecessary or inappropriately placed information. This will only adversely affect software usability. Some of the information in that manual should have been dropped or covered in another appropriate information element.

As described in Chapter 1, the usability components of a software system are user objects, user actions, interaction information, and support information. Gould and Lewis [3] say, “Unfortunately, these interacting pieces are usually designed separately. Definers, designers, implementers, application writers, and manual writers constitute large groups in themselves, and are often separated by geography or organization. . . . It appears that superior quality can be attained only when the entire user interface, including software, manuals, etc., can be designed by a single group, in a way that reflects users’ needs, and then evaluated and tuned as an integrated whole.”

What’s critical, therefore, is integration. Integration is the software usability-driven approach to identifying all the information elements users need and expect, and designing, evaluating, and improving them as an integrated whole. In this definition, “identifying” implies deciding on the content, media, and form that users need and expect from each information element. And, “all the information elements” covers labels, messages, online support elements, and printed support elements.

Integration might appear obvious. But, do software houses integrate interaction and support information? Do they even integrate printed and online support elements? Here is what technical communicators have expressed about the process of developing printed and online information [3]:

- *Online documentation should be written after printed documentation.* The technical communicators' reasoning is that the online document is a subset of printed information. They also say that printed information contains many different structures such as introduction, procedures, examples, glossary, and reference. The problem with this approach is that online information is not designed specifically for online use. For example, online text taken from a printed user's guide may not provide the quick how-to information users expect while working online.
- *Online and printed information should be written independently.* The technical communicators say that the same text should not be used to serve different purposes. Therefore style, formatting, and retrieval techniques for online text will be different. Even in big companies, online Help, for instance, is often planned and written separate from the printed manual—often in another city. The problems with this approach are as follows:

Information provided in, say, online Help is repeated in, say, a printed manual when there is clearly no such requirement from the users' viewpoint.

Worse, if the Help (or manual) writer incorrectly assumes that the writer for the other media will cover certain topics, critical information could be left out.

Error message might say that how-to information is available by pressing F1. When users press F1, all they get is what-went-wrong information.

- *Printed and online information should be generated simultaneously from a common database.* The technical communicators say that printed and online documents contain the same information and that this ensures consistency.

Here are the reasons why you should apply the integrated approach:

- Ensures that users have all the information they require—in the media and form they need or expect;

- Eliminates information users do not need or expect;
- Eliminates unnecessary repetition of information (note that repetition may be required from the user's viewpoint);
- Eliminates user dissatisfaction resulting from the above;
- Changes the user perception that using support information is an unnecessary diversion;
- Often results in smaller information elements with improved retrievability.

Because of interdisciplinary collaboration in UCID, when you take the integrated approach within a user-centered design process, you will have integrated information elements and, more important, the information elements will be integrated with the rest of the software usability components. Users will perceive the overall software system as one single usable system.

7.3 Information architecture: design process

The information architecture should be designed by the technical writing manager or senior technical writers in consultation with the usability engineer and the programming team. The initial information architecture is designed during the UCID high-level design phase. It should be improved based on task (user interface) complexity data, which may be available later in the project.

Integration clearly is the way to maximize software usability. Unfortunately, there is no single right mix of information elements—a predetermined set—that you can provide for every software system. For example, among the three software products of Netscape Enterprise Server, Microsoft Word, and IBM Interactive Storage Management Facility, everything is different. The user interface is different. User groups are different. Information requirements are different. Obviously, the mix of information elements for each of these products is

different. Similarly, you should determine the specific information elements required to support the information needs and expectations of your product users.

Nor can we define a comprehensive “IF...THEN...” formula. For example, I cannot say, “If you provide online Help for error messages, you don’t need to provide a Messages section in the printed guide” because that depends on many factors that only the project technical writers know.

Therefore, what I provide here is only a framework to get you started on the right track.

The question you are trying to answer is, “What information elements should be provided?” In other words, what sets of topics should be separately packaged for which media? To answer this question, you should know the following:

1. *Users’ information requirements.* You should know users, tasks, and task (user interface) complexity.
2. *Media characteristics.* You should know which media is best for what type of information and what user situation.
3. *Project-specific requirements.* You should be aware of things like resource and schedule limitations that are specific to the project.

Armed with that knowledge, you should do the following:

1. Visualize the use of the software system to determine what information users will need and *when*.
2. Pack a set of topics into each information element. For this, you should also consider various packaging approaches, such as task-orientation described later in this chapter.
3. Title each information element.
4. Document the information architecture. List all the information elements that you should provide, describe what each element contains, and show how each element fits into the whole. You can place these details in the UCID plan.

7.4 Know information requirements

Information elements provided should contain just the information users need and expect. Exclusion of some information may affect the users' task performance. Presenting information that is not required is a waste of time and effort. Moreover, it worsens retrievability and makes manuals bulky.

To find what information users need, you need to know the users, tasks they perform, and the complexity of the user interface.

7.4.1 Users and their tasks

The more you know about users and tasks, the more you will know about the information required. See Chapter 5, which is devoted to this topic.

One of the things you should do to determine information requirements is construct a matrix, where you list the tasks against the user groups that will perform the tasks. The simple user/task matrix in Table 7.1 indicates the possibility of each user group performing a task. The ranking is on a 0 to 5 scale, where 0 indicates that there is no possibility. Use this matrix to identify the most important user-software interactions.

In another user/task matrix (Table 7.2), you could show which user group will use which information elements to perform what tasks. It might be useful to summarize here the results of a survey [4]. The survey asked experienced users and writers of online information to rank their preference of four "instructional" elements on a scale of 1 (most preferred) to 6 (least preferred). Here are the results:

Table 7.1

Matrix Showing the Possibility of a User Group Performing a Task

User/Task	Clerks	System Administrators
Installation	2	5
End Use	5	0

Table 7.2

Matrix Showing Who Uses Which Elements to Perform What Tasks

User/Task	Clerks	System Administrators
Installation	Labels, messages, user's guide	Labels, messages
End Use	Labels, messages, online Help, user's guide	None

- online tutorial (average 2.58);
- classroom course (average 2.65);
- hardcopy tutorial (average 3.0);
- trial and error (average 4.0).

7.4.2 Task (or user interface) complexity

This is about how users will perform the tasks as implemented in a particular software system. Knowing task complexity is an excellent way to know what types of information users will need and *when*. You will, however, need at least some idea of the look and feel of the user interface, which is unlikely to be available at the start of the UCID high-level design phase. Iteration of the information architecture should therefore continue, at least until some user interface is available.

To understand task complexity, you could try a technique described by Doris [5]. Using a list of 10 questions, Doris set up a difficulty index to describe the tasks users can perform with a software system.

1. Does this task require a single key response?
2. Does this task require the use of more than one key?
3. Does this response require synchronized operation of two or more keys?
4. Does this task response consist of a sequence of keystrokes?
5. Will users need to reposition the cursor?

6. Does this task require numbered steps to explain it?
7. Does this task response result in one or more new screens?
8. Does this task require a typed response?
9. Does this task require a choice from a menu?
10. Does completion of this task leave users on a new screen?

An analysis based on a difficulty index can indicate user interface complexity. In the context of the example software described, Doris remarks that the easiest task is “Deleting a character to the left,” which required only one keystroke. The most difficult task is “Selecting a specific page to print,” which requires the use of five key responses, involves three or more steps, results in two new screens, requires two typed responses and three menu choices, and results in users being left on a new screen.

Such indications of user interface complexity help you in designing the information architecture as well as in writing the information elements.

7.5 Know media characteristics

The medium of information delivery is a decision you make at the time of designing the information architecture. To make that decision, you must first understand the particular media where users will need or expect certain types of information. You should also know what kind of information would better go online and what would better remain on paper—from the users’ viewpoint.

Print and *online* are the media most often used for communicating information. In the following sections, we will look at the effectiveness of online and print media in terms of retrievability, usability, and readability.

The *World Wide Web* is a new media for providing support information. Although information appears on the users’ screens, I’m not discussing Web support information under online media because of the Web’s own unique characteristics. Today, you should consider Web-based information elements for your software products. The possibilities are enormous, including multimedia capabilities and pointers to useful

Web sites and newsgroups. Cost savings are possible for the software house that cuts down on printed information. Disadvantages with Web support information are the need for Internet access and slower access to information.

Video is another support information media. It is primarily used for presale demonstration and tutorials. The advantage is that it is fairly well suited for demonstration. The disadvantage is that it demands a TV and VCR.

The *audio (tape)* media is used, but rarely. Its obvious advantage over online media is the absence of screen clutter! The disadvantage is that it demands a cassette player.

7.5.1 The print media

Even after the “invasion” of the online media, print media is still used for its many advantages. Examples of printed information elements are user’s guides, reference cards, and instruction foldouts.

Types of information better suited for the print media include information that:

- Is “a lot of text;”
- Contains concepts;
- Demands careful or detailed reading;
- Must be available when a computer is unavailable (e.g., installation and crash recovery information);
- Users may want to take home;
- Is legal, such as copyright, disclaimer, license, warranty, and safety information.

7.5.1.1 How effective is print media

How good is print media? Let’s review it in terms of retrievability, usability, and readability attributes.

Retrievability Compared to the online media, retrievability with print is generally slower. However, there are many techniques you can use to

maximize print retrievability. Chapter 13 describes retrievability techniques, which you can consider for your printed information elements.

Usability Printed support elements are difficult to make available to everyone. Often, users need to walk up to the bookshelf or library. Printed elements can get loaned, misplaced, or lost. They take up desk space. One advantage is that most users find it easy to perform tasks online as they read instructions from a printed element. Finally, there is the advantage of portability.

Readability If designed carefully, printed information elements can have excellent readability. Here are certain factors that contribute. Resolution is excellent, with typeset pages having a typical resolution of 1,200 to 2,400 dots per inch. You can also use good graphic design to enhance readability. Size is a contributing factor. Size varies, but is usually small enough so that two full pages can be seen in one glance. Shape is another factor. It is predominantly portrait.

From the software house's viewpoint, one disadvantage with print is the production process it demands. A separate, time-consuming, and expensive production process is required through an external agency.

7.5.2 The online media

The online media is the computer screen, where support information is displayed. The information element is often “physically” part of the software system itself. Information better suited for online media include quick “how to” information. Examples of online information elements are Help, tutorial, and cue cards.

7.5.2.1 How effective is online media?

At information design seminars I give, when I ask participants which media is better, most of them would put up their hands for online. But when I get down to specific questions such as, “Is it easier to read on the screen?” they start thinking again. In fact, there are people—like me—who would rather *print* it and read. Just like print, online has its advantages and disadvantages.

Retrievability If designed carefully, online retrievability is clearly superior to print retrievability. Users can retrieve and display information with a keystroke. The best thing about online retrieval is that even if you have a large amount of information, users do not perceive going from one “manual” to another. Hypertext and other techniques make retrieval a breeze.

Usability One of the best things about online media is the fact that users perceive online information as the software system explaining itself. In contrast, a printed manual is physically separate from the system and, therefore, users might think of it as a hindrance to the task they are performing with the system.

The online ability to display context-sensitive information has no parallel in the print media. Users can get just that piece of information they need at the moment—with one keystroke. Depending on the user’s task context and his or her earlier performance, you can even automatically display contextual information.

Users do not perceive the “bulkiness” of online information. Only what they request appears on their screen. Moreover, removing the information is usually quick and easy.

Users, however, have complained about online usability in terms of them having to turn information “off and on” to continue using the instructions. This problem is eliminated in many new software systems, where a small window containing the instruction always stays while users read and apply the information to their application task.

Bad design can create problems. If the user interface for an online support element (e.g. Help) is significantly different from that of the user interface for the software system, users have to learn how to use that element. This defeats the purpose of providing Help or other online support elements. Bad design can also cover or clutter users’ working screens.

Readability Studies have shown that online readability is a loser against print readability. Resolution, the spacing between pixels on the screen, is not yet as good as on a printed page. Moreover, users cannot easily adjust the reading distance. Whereas most paper-based graphics can be shown on the screen, not everyone’s computer will have the required resources, such as high-resolution monitors. The bottom line is that on-screen

reading is 20% to 30% slower. For many users, online reading can result in eyestrain and headache. Online readability may be improved in the future when technology advances and become affordable.

7.5.2.2 Why developers think online is better

Here are some reasons why developers prefer developing information for the online medium.

Easier to produce and distribute For producing online support elements, you don't have to use an external agency, whereas for printed support elements, you will go through a print shop. Of course, online information elements have to be put on a diskette or CD-ROM. However, this duplication usually happens alongside—and the same way as—the rest of the software system. In general, online information is also less expensive to duplicate, store, and distribute.

Better acceptance Software engineers generally accept online technical writers more readily as fellow team members. Online writers themselves accept and enjoy the technical writing profession better because online information is “physically” part of the software system and because of the various technical procedures, such as testing, that is involved.

7.6 Know project-specific requirements

We may have a design that is excellent from the users' viewpoint. From a business viewpoint, however, the solution may be infeasible. It is important that we strike a balance between an excellent solution and an infeasible solution by understanding the realities of the business environment in which the software engineering project exists. Some of the business issues you should analyze are as follows:

- *Skills*. For example, is creative talent available for animation design?
- *Tools*. Are tools available for prototyping an online tutorial?

- *Cost.* Will this expensive tool pay off in the long term? Is the budget too low to produce a complex set of information elements?
- *Schedule.* Software houses are always looking for ways to reduce time to market. Is there time to learn the new tool?

7.7 Package information elements

Decide what will go into labels. Also, decide what types of information will go into different message types, such as informative messages and warning messages. For packaging of support information elements, consider the following approaches.

7.7.1 Packaging approaches

Three standard ways of packaging information are described here. They are task-orientation, role-orientation, and software-orientation. Books that describe these approaches recommend task-orientation alone. However, you should consider all three approaches to see which best meet users' information requirements. Note that here we are only talking about high-level packaging of information into elements. Low-level design and writing should always be task-oriented, irrespective of the packaging approach you implement. This is simply because users are performing *tasks* with the software.

Task orientation The information to be provided is based on an analysis of the use of the software and is limited to what users need to perform specific end-use and support tasks. Information that is not required to perform tasks is useless information and is not provided.

Role orientation If you use this approach, you will have each information element targeted at each user group. For example, you might have a system administrator's guide, a user's guide, and so on. You should consider this approach if roles such as system administrator are clearly defined in your customer organizations.

Software orientation If you use this approach, you will follow the software system's structure consisting of modules or functions. Here, you might have an information element each for a major function of the software system.

7.7.2 Some tips to consider

Here are some tips you should consider:

- Consider size. Avoid packaging a printed element of over 300 pages or less than 20 pages. A large reference is not too bad because users are just looking for a specific topic (in contrast, users may read an entire chapter of a guide).
- Keep all guide information together, but not necessarily in an exclusive information element. Depending on expected size and use, you can put guide and reference information together in a single information element.
- Avoid unnecessary repetition. If you want three manuals, all of which have, say, over 30 pages of common information, put that information in a single common manual.
- Avoid requiring users to jump from one information element to another.
- For mainframe software systems, users may be clearly defined as system administrators, operators, and so on. Here, you probably should not put system administration and operation information together in a single manual—especially if users have the option to order manuals.
- If your software system has a function that lets users make extensive changes to the software's operation, you may want to separately place the description of that function so that it is only available to "privileged" users.

7.7.3 Give it a title

The goal of the title for an information element is improved retrievability. The title can include some or all of these:

- Product name;
- Content;
- Approach;
- Users;
- Purpose;
- Company name.

Many software houses have their own rules for titling manuals. IBM mainframe manuals typically have the structure shown in Figure 7.2.

Normally, companies strive for consistency in titling. However, it is more important to evaluate titles for their retrievability.

7.8 Information architecture: the OS/2 Warp example

The example given here is a set of online and printed support information elements for an IBM product. It does not cover the interaction information.

The product's printed information element, *User's Guide to OS/2 Warp* [6], provides an interesting road map to the online information elements users may use. Let us first take a look at the printed element itself.

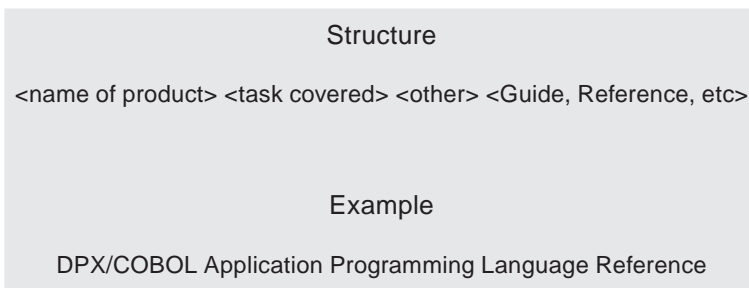


Figure 7.2 Titling in mainframe manuals.

7.8.1 User's guide

The printed information element, *User's Guide to OS/2 Warp*, has four parts and three appendixes.

The four parts are:

- Part 1—Installing OS/2.
- Part 2—Exploring OS/2.
- Part 3—Using OS/2.
- Part 4—Troubleshooting (Here, you will find an explanation and action for each error message that can appear when users install the product).

The three appendixes are:

- Appendix A. Keyboard and Mouse Use.
- Appendix B. Books for OS/2 Version 3 Application Developers (Here, you will find brief descriptions of a large number of printed manuals that users who develop applications for OS/2 can order).
- Appendix C. Notices.

7.8.2 Online Help

Online Help is available for every desktop object, every menu item, and wherever there is a Help push button.

- *F1 Help*: Provides information about objects, pop-up menu items, entry fields, and push buttons.
- *Help push button*: Describes most of the object pop-up menus and notebook pages.

7.8.3 Online books

During installation of the product, the online information and books shown in Table 7.3 are added to the Information folder.

Table 7.3
Information Elements for IBM OS/2 Warp

Information Element	Description
Master Help Index	Provides help for almost everything the user may want to know about the OS/2
Command Reference	Describes all the commands that can be used at the OS/2 command prompt
Glossary	Defines the terms used in various OS/2 information elements
REXX Information	Helps users get familiar with REXX language and programming concepts
Windows Programs in OS/2	Explains how to use OS/2 and Windows programs together under OS/2
Application Considerations	Explains how to enable certain programs to run under OS/2
Performance Considerations	Provides information about system performance improvement, memory management, communication ports, and so on
Printing in OS/2	Explains everything from installing a printer to solving printer problems
Multimedia	Describes multimedia programs available with OS/2
Trademarks	Describes trademarks mentioned in various online information elements

References

- [1] Constantine, Larry L., "Toward Usable Interfaces: Bringing Users and User Perspectives into Design," *American Programmer*, Feb. 1991.
- [2] Rubin, Jeffery, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, New York, NY: John Wiley & Sons, Inc., 1994.
- [3] Gould, J. D., and Lewis, C., "Designing for Usability: Key Principles and What Designers Think," *Communication of the ACM*, Vol. 28, No. 3, March 1985.

- [4] Brockmann, John R., *Writing Better Computer User Documentation: From Paper to Online*, New York, NY: John Wiley & Sons, Inc., 1992.
- [5] Watts, Doris R., "Creating an Essential Manual: An Experiment in Prototyping and Task Analysis," *IEEE Transactions on Professional Communication*, Vol. 33, No. 1, March 1990.
- [6] *User's Guide to OS/2 Warp*, IBM Corporation, 1994.