5

DEFINE PROTOTYPE CONTENT AND FIDELITY

MANAGER'S CORNER

DEFINE PROTOTYPE CONTENT AND FIDELITY

Defining prototype content and fidelity determines how much time and resources need to be spent on the prototype. A prototype with too high a fidelity wastes resources by spending time creating a level of detail that will need to be done all over again in later iterations. A prototype with too low a fidelity results in not getting enough information to validate requirements and assumptions. You select the fidelity by choosing the content and setting its fidelity. In choosing the right prototyping content and setting its fidelity correctly, it is important for a manager to consider the following points.

Beware of Too Much Content With Too High Fidelity

Well-meaning employees want to impress everyone. Unfortunately, this usually results in creating a prototype with too much high-fidelity content; because high-fidelity content is more impressive looking than low-fidelity content. When reviewing an early high-fidelity prototype, ask yourself how much are real design decisions and how much are just unnecessary details.

It Is Better to Underestimate the Content and Fidelity Than to Overestimate It

It is best to start off conservative with content fidelity levels. This requires less work and enables you to iterate the prototype to the right level of fidelity. Iteration allows you to progressively build on previous work, which prevents throwing a prototype away. For example, if you prototype with too low fidelity, it still sets a general direction to iterate on. Once the direction is decided, you can iterate and build on a direction to a higher fidelity. Once a big direction change is made in the requirements or as assumptions are invalidated, all related design work needs to be revisited if not redone completely. This commonly happens when the fidelity is too high, and usability testing or some other validation proves an assumption wrong. The work must be repeated and valuable time is lost. Therefore, it is important to get the low-fidelity stuff right first and then work out the higher fidelity details.

Fidelity Too High or Too Late

There are two fundamental rules regarding high-fidelity prototypes:

1. Prototyping best practice dictates that you should move from low to progressively higher fidelity prototypes through the design process. Early high fidelity prototypes are throw-away proof of concepts.

2. High fidelity requires a conceptual model. The more detailed the design, the more it will rely on a conceptual design to make it fit together. A conceptual model is an overall design framework that is the design rationale for things such as metaphors, language usage, and terminology. It is how the designer believes the user will conceptualize the software (e.g., desktop software as a desk metaphor). The user in turn then interprets the clues from the conceptual model into their own mental model.

The task flow in Chapter 4 established how the prototype should *work*, whereas this chapter establishes how the prototype should *look*. A prototype can contain many different kinds of content, including editorial, visuals, and navigation structure. In general, you should not prototype everything at once, especially in the beginning of a project. Even with a mature product, it may be wise, to isolate prototype content destined for revision. This chapter will help you discern between important and trivial content and decide what to include in your prototype.

THE TWO MEANINGS OF HIGH FIDELITY

high fidelity 1. A kind of sound-reproducing system whose realism of reproduction is judged to be better than average. Stereo reproduction can be high fidelity or otherwise. 2. The pursuit of perfection in sound reproduction, as a hobby or a religion (from Stereophile.com, accessed January 18, 2006).

We bring this definition to your attention to make sure you avoid falling into the high-fidelity religion. High fidelity always sounds more impressive than low fidelity. However, in prototyping, unlike sound reproduction, you want to build up to high fidelity and not just plunge in the high-fidelity religion that "a sexy looking prototype is always the best."

PROTOTYPE FIDELITY

Fidelity is the level of detail that content is rendered in the interface. Prototype content can be rendered in a continuum from lowest to highest fidelity. Specifically, the fidelity of visual look, the interaction behaviors, navigation flow, and other aspects of the user experience as reflected by prototyping content.

A high-fidelity prototyped content is more like the actual released software. For example, with the highest-fidelity visual design the software *looks* like it will when it is actually built. With high-fidelity interaction design the prototype *behaves* like the released software would. Therefore, it is possible for a prototype to have variable levels of fidelity for the different prototype contents.

Why do you want differing level of fidelity?

You might choose to do this to focus just on a new visual design or a new interaction concept, etc. By deliberately making some elements high fidelity, the audience is better able to focus on the higher fidelity items, giving them an unequal weight and thereby the lead focus. The most common prototyping mistake is reflected in the second Stereophile.com definition: when high fidelity becomes a religion. When usability testing is relegated with an almost religious fervor to extremely high-fidelity prototyping, you'll almost surely be testing too much to receive valuable undistracted feedback. In our experience, waiting to prototype only in high fidelity once all research and design is completed is to prototype too late in the game. Prototyping can uncover major problems. So it is best to start early before it is too late. Start with prototyping in lower fidelity and iterate toward higher fidelity.

Contrarily, when you want to just try out a concept, you may decide to keep the concept in lower fidelity, because the lower fidelity keeps unnecessary details from distracting the software-making team. In the early stages of design, low-fidelity prototypes are most appropriate for allowing designers to evaluate and try multiple designs, because they appropriately constrain the focus of evaluation and trial to the larger more conceptual aspects of software design–the user experience in terms of sequencing and flow.

As a design moves incrementally closer to development and its details start to fall into place, different prototyping methods are used to portray the increasing detailed solution of the software. The fidelity representations can range from low (rough hand-drawn sketches on paper) to medium (digital wireframe representations) to high (detailed renderings of visual user interfaces in Photoshop) to highest (a beta version of the actual software).

Figure 5.1 shows a low-fidelity prototype, a rough sketch in which the interaction can be approximated. There is no attention paid to a high-fidelity graphic design or even the information displayed. Figure 5.2 shows a high-fidelity visual design prototype that looks similar to the intended actual released software. It looks like what you'd expect to see on a computer screen, and as a high-fidelity prototype, it also behaves like finished software (though this complete behavior is not required for a prototype to be considered high fidelity). Some high-fidelity prototypes may only be high fidelity in one or two content aspects. More on this below.

Low Fidelity

Low-fidelity prototyping content is akin to sketches that artists and designers create early in the ideation stages of a design concept: vague and minimally formed in attributes. Low-fidelity prototypes create an early representation of its content. Low fidelity allows designers early conceptualization of page layouts, such as the rough positioning of menu areas, banners, toolbars, and content areas. One of the primary benefits of low-fidelity prototypes are usually rapid to produce. So you generally want to make your biggest mistakes with low-fidelity prototypes because of the lower cost of iterating them.

FIGURE 5.1 *Low-fidelity prototype.*

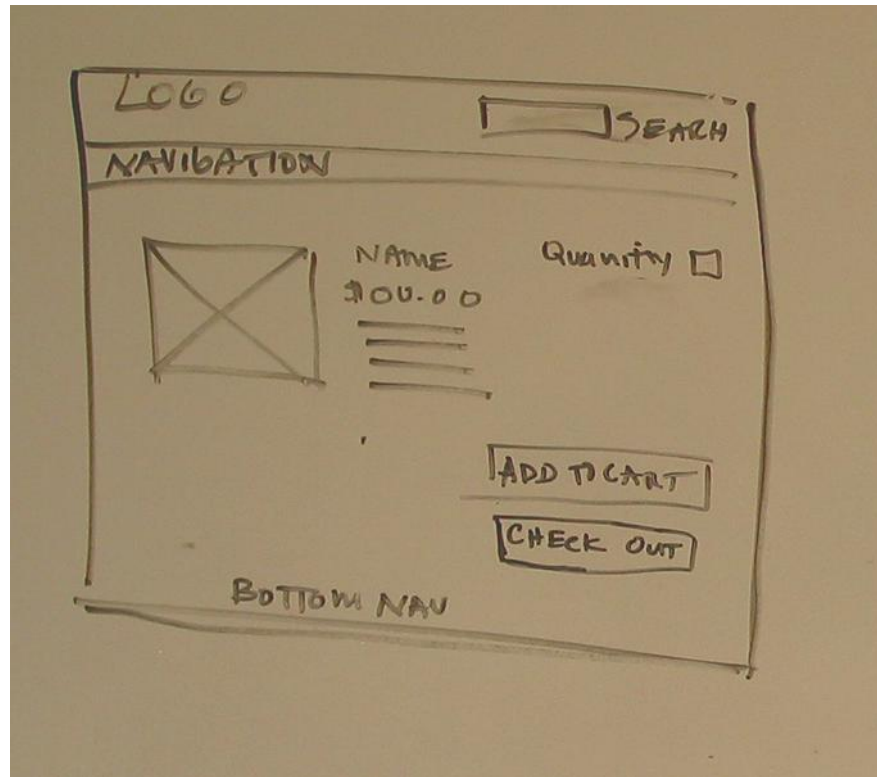


FIGURE 5.2 *High-fidelity prototype.*

High Fidelity

People often mistake high-fidelity prototypes for the finished products. This dilemma of *demo ware* can best be surmised by the classic example of Apple Computer's "Knowledge Navigator" video prototype. Knowledge Navigator, a futuristic vision statement, was so realistically portrayed in a video scenario that the consuming public thought it was an immediately available product and made inquiries about purchasing it.

High-fidelity prototyping content has a more accurate look and/or feel to the final product. A high-fidelity prototype, like an artist or designer's comp, is rich in detail with all its attributes. High-fidelity prototypes (prototypes of mostly high-fidelity content) are intended for designers to create and try out the contents of a user interface, content such as information design, visual design, or interaction design. Ideally, the purpose of a high-fidelity prototype is to test the content with end users or at least to get their direct feedback using some other mechanism. High-fidelity prototypes also allow the designer to efficiently document the design without lengthy documents that are open to misinterpretation. Because high-fidelity prototypes are usually longer to produce, you generally want to have completed the bulk of usability testing before you create a high-fidelity prototype. Because of the high cost of iterating these, you want to refine the prototype at this stage, not make wholesale changes.

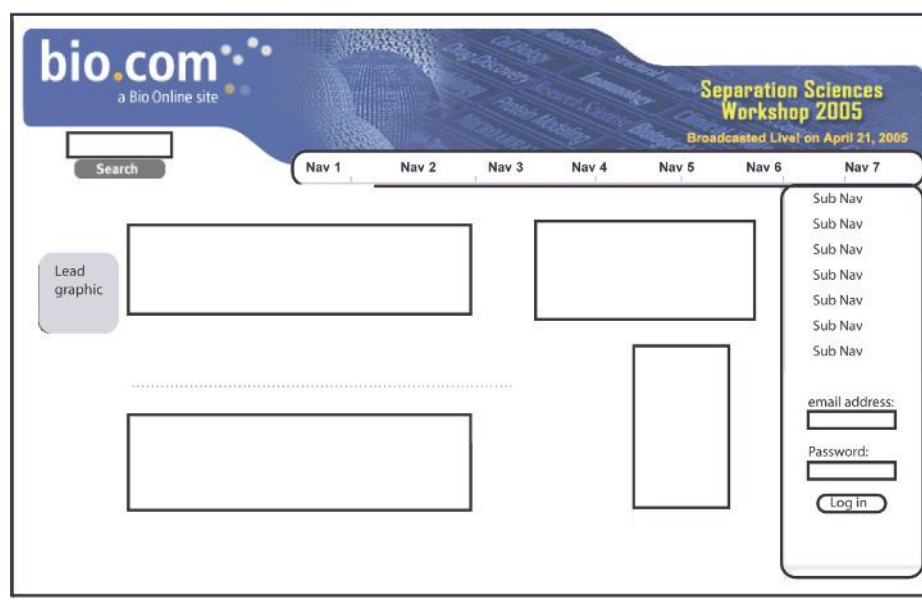
PROTOTYPE CONTENT

As the case study later in this chapter illustrates, you can avoid miscommunication and misunderstanding over what a prototype is trying to show by judiciously choosing the correct content and fidelity level for the prototype. The prototype context needs to be established by including all the different forms of content, but content can be emphasized or deemphasized through the interplay of high and low fidelity to prevent items from unwanted focus when the prototype concept is presented. Figure 5.3 shows a typical experiential prototype in which all the contents are clearly visible. Figure 5.4 shows the opposite, a prototype in which almost all the content is masked or deemphasized (low fidelity), forcing the viewer to concentrate only on the overall structure. Certain prototypes are chosen to emphasize certain content fidelities over others to evaluate them. For example, a wireframe prototype is typically much lower in fidelity than a prototype programmed in code; however, it is enough to evaluate a visual design direction, whereas a coded prototype would mean involving not just the visual design but also the interaction design, system performance, and so on, thereby detracting from the visual design. Moreover, if given the stage in software creation you're only prepared for visual design, by using a coded prototype only the visual

design will be ready for prime time; the rest–interaction design information and so on–will all be placeholders. During evaluation, the stakeholders may get sidetracked by the information or interaction design and either ignore the visual design or, worse, judge the visual design poorly as not having been well thought out. Therefore, using the right level of fidelity at the right time is essential for effective prototyping.

FIGURE 5.3 *All content is visible in this prototype.*

FIGURE 5.4 *In this prototype screen, all content is masked except for branding elements.*



Fidelity is just one characteristic of prototyping; the others are discussed in Chapter 6. For now, we concentrate on the fidelity of different kinds of content including, but not limited to:

- Information design
- Interaction design and navigation model
- Visual design
- Editorial content
- Brand expression
- System performance and behavior

Each content category plays a different role in prototyping. Different techniques can be used for emphasizing (heightening the fidelity) or deemphasizing (lowering the fidelity) the visual and interactive components of prototypes. Given the connection of content to fidelity, think of a prototype in terms of the content/fidelity matrix shown in Table 5.1.

TABLE 5.1 Content Fidelity Matrix

Content	Very Low Fidelity	Low Fidelity	Medium Fidelity	High Fidelity	Highest Fidelity
Information design					
Interaction design					
Visual design					
Editorial content					
Branding expression					
System performance					

Information Design

Information design is the design and structure of information used in an interactive piece of software, including information in the form of data entry and display fields, menus, tables, graphics, messages, and other expressions of information. Information design by this understanding can also mean information architecture. Information architecture has a slightly narrower meaning [Rosenfeld and Morville 2002]:

- The combination of organization, labeling, and navigation schemes within an information system.
- The structural design of an information space to facilitate task completion and intuitive access to content.
- The art and science of structuring and classifying websites and intranets to help people find and manage information.

For our purposes, information architecture is the systematic organization, flow, and communicativeness of text-based editorial content in the user interface (see

Editorial Content, later in this chapter). Information design, on the other hand, separates textual information content (what we call editorial content) from content structure; this structure should be seen in combination with the syntactic, semantic, and pragmatic expressions of icons and other visual elements.

Because it embodies the organization, flow, and expression of screen content, information design can be the most powerful element in the user experience. The visual rendering in combination with the editorial expression of information often sets a tone–a feeling of friendliness or aloofness. Information design, by using text, can also be instructional and functionally explicit. All too often users get confused and can't find key functionality available on their screens, simply because the wrong terminology is used or the information is located outside the user's direct field of vision. Especially for information-intensive software systems, such as websites and services, providing the right information organization, placement, and editorial communicativeness is of utmost importance. Via carefully considered and well-crafted information design, designers can literally identify and speak the visual and editorial language of targeted users.

Techniques to Adjust the Fidelity of Information Design

In prototyping, information design is emphasized (higher fidelity) through more detailed visual representation of the high-level textual structure and the navigation flow (see Figure 5.5). The three most common prototyping methods used to focus on information design in software are card sorting, wireframe prototyping, and high-fidelity experiential prototyping. The easiest way to focus on the information design is to simply exclude other prototype content so only the structure or more detailed representation of information is shown. As shown in Figure 5.6, you can also focus on information design by deemphasizing the graphic elements and even the body text in a wireframe prototype, only displaying the menus, headings, buttons, and other navigational items. The most thorough and complete method to emphasize information design is by using an experiential interactive prototype (paper or digital) where all finalized information is displayed. This method is the most explicit in terms of the syntactic connection between the information and the information design that supports it. However, building an interactive experiential prototype is time consuming and makes more sense for final design refinements rather than for early prototyping.

In general, when developing the information design, you can lower the fidelity whenever a prototype is a narrative and the audience is not directly engaged with the prototype. The most common way to deemphasize information design is by using color blocks in the prototype as a placeholder for text (lowest fidelity). Selectively blocking out text allows the designer to include some incomplete elements of the information architecture. It also allows the user to block out controversial items to avoid them overtaking a design discussion, as in the example at the beginning of this chapter. One step up from blocking is the use of greeked text (low fidelity). Greeked text is purposefully unintelligible but allows the audience to see the visual design of the typography.

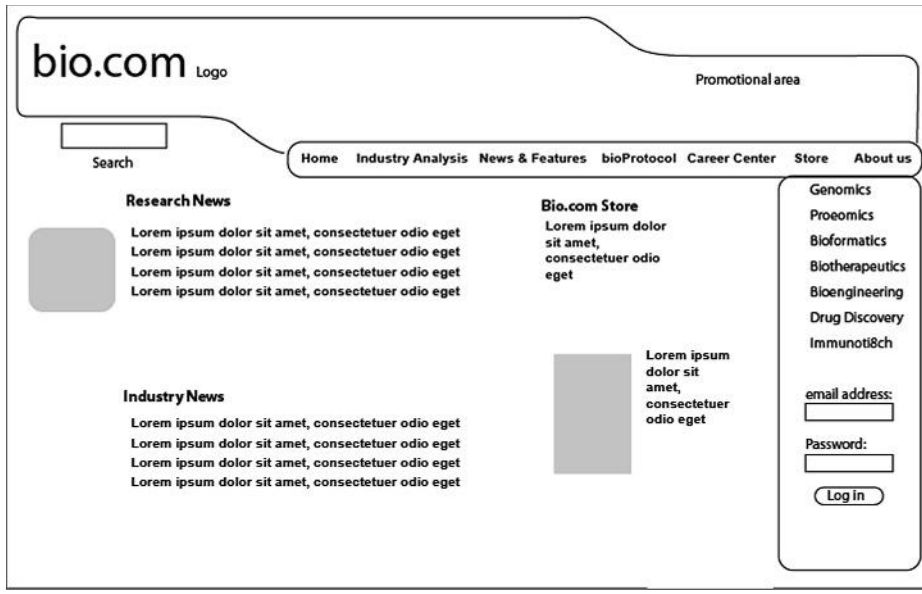


FIGURE 5.5 *Information design in high fidelity in a prototype.*

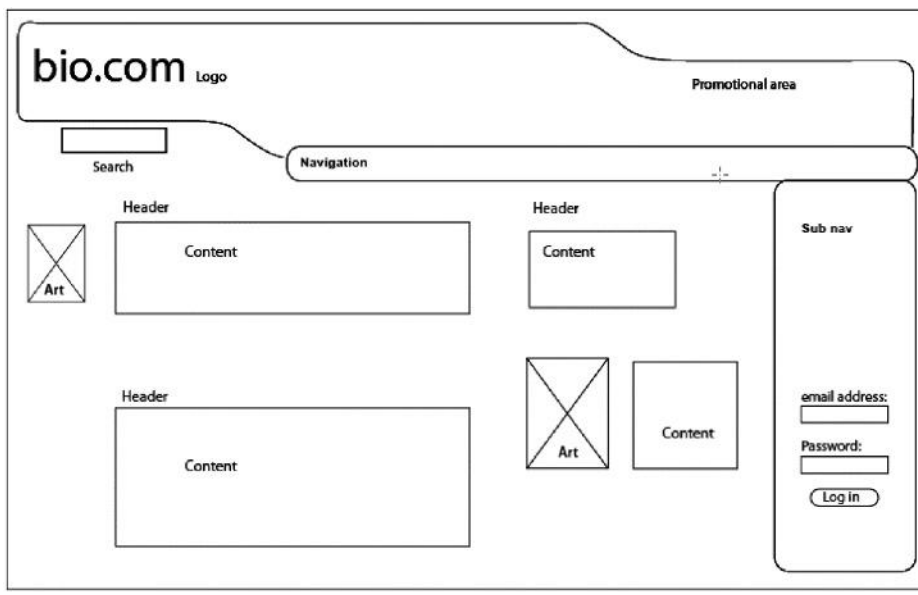


FIGURE 5.6 *Information design in low fidelity in a prototype. The only thing clear about the information design from this prototype is its raw structure.*

INTERACTION DESIGN AND NAVIGATION MODEL

Interaction design is how the user interacts with software both in the structure and the flow within a screen and among screens. The design determines how the system responds to user actions. If the user navigates to another window, is it a window where a primary or secondary task occurs? How does the system respond? Does it allow a task to be completed, provide information about errors, send warning messages, or provide processing status? Can the software

anticipate a user's next action or must users always initiate their next step? These are all factors in the interaction design.

The interaction designer owns the interaction design. As with any aspect of software usability, interaction design content comes from understanding user tasks, human behavior, and testing designs with target users. The art of interaction design comes from taking a complex and often contradictory array of user interaction needs and developing a single usable design. Information designers, interaction designers structure their interfaces so that a single consistent interaction model with redundant command structures which can support a variety of users and their different styles of usage.

For an interaction designer, nothing beats the comfort and satisfaction experienced by a user who can effortlessly engage in and complete an activity. Likewise, nothing is a bigger waste then creating functionality a user cannot use or even find.

Techniques to Adjust the Fidelity of Interaction Design and Navigation Model

Interaction design and navigation are higher fidelity whenever the points of interaction and navigation flow can be portrayed in their most polished and detailed form: every link, command, and function as embodied by a user interface element is represented and their resulting action/navigation is specified. This high fidelity can be achieved in many ways, but two are the most common. One is through the level of complete narration and the other is through the level of complete interaction. Complete narration is when a storyboard includes not just a sunny day interaction but also all possible exceptions and alternate interactions and navigations. Complete interaction is when an interactive prototype has every working command, user interface controls, links, and so on, either working or at least specifying what will happen when one uses them as well as how one uses them.

One last technique for deemphasizing the interaction design is to make the prototype interactive in a way that is irrelevant for the ultimate interaction of the software. A good example of this type of prototype is a card sort, where the user interacts intimately with the prototype but in a way that has nothing to do with the ultimate user interaction of the software. In fact, card sorting is a method used to not only understand how users perceive an information structure but also how they understand an interaction scheme in the context of the information structure.

VISUAL DESIGN

Visual design is the visual language and composition of visual elements used to express the software structure and messages to users. Finding the right balance between visual structures and messaging is a complex iterative design activity. The interaction designer will tell you that the structure and message should express the use of the software. The information designer believes that the structure and messaging are meant to lead the user to desired information. The branding specialist

will inform you that the structure and messaging should express the values, attributes, and qualities of a company, product, or service. As you can tell from these various purposes, visual design is key in establishing users' mental models through metaphors, analogies, concepts, and other mental associations.

The primary considerations of visual design include [Dondis 1973]:

- Layout (visual composition)
- Typographic design
- Graphic elements, including geometric/non-geometric shapes, icons, buttons, photographic images, illustrations, and animations
- Fonts selection
- Color scheme
- Visual branding expression

Because it is the most tangible piece that people can immediately grasp and relate to, most users think of the user experience purely in terms of the visual expression. Furthermore, visual elements can draw the most powerful emotions and reactions from users. The attributes of visual design make it a powerful element of the user interface and of a prototype.

The owner of the visual design is the visual designer. Visual design is only partially informed by user research. Visual design is also informed by competitive analysis, market trends, organization needs, executives' spouses, and that cool thing the designer saw when driving to work.

Techniques to Adjust the Fidelity of Visual Design

In general, the fidelity of the visual design is heightened by a color representation of the software with all graphic elements (widgets, logos, typography, color usage, etc.) (see Figure 5.7). Visual design prototypes are usually digital (or color prints of a digital interface) and are typically used late in the design process, once the interaction or information needs are firmly established (or when the prototype is meant to serve as a presentation demo).

As shown in Figure 5.8 fidelity of the visual design is low when a prototype either does not show the graphic user interface, such as in a narrative scenario prototype, or when an abstracted prototype displays just one or two design aspects, such as rough sketch wireframes and card sort prototypes. Prototypes that deemphasize the visual design by representing the user interface as grayscale sketches devoid of detailed graphic elements. The sketches often block out logos and use just text buttons instead of icons, making the designs purely textual interfaces. Sometimes designers try to suggest a visual design in a wireframe by including a company logo or using typography in a certain way. However, these are meant to be placeholders and information for designers and usability test participants to consider rather than an actual representation of the final visual design. By deemphasizing the visual design allows the audience to concentrate on just the information or interaction design.

FIGURE 5.7 *Visual design emphasized in a prototype.*

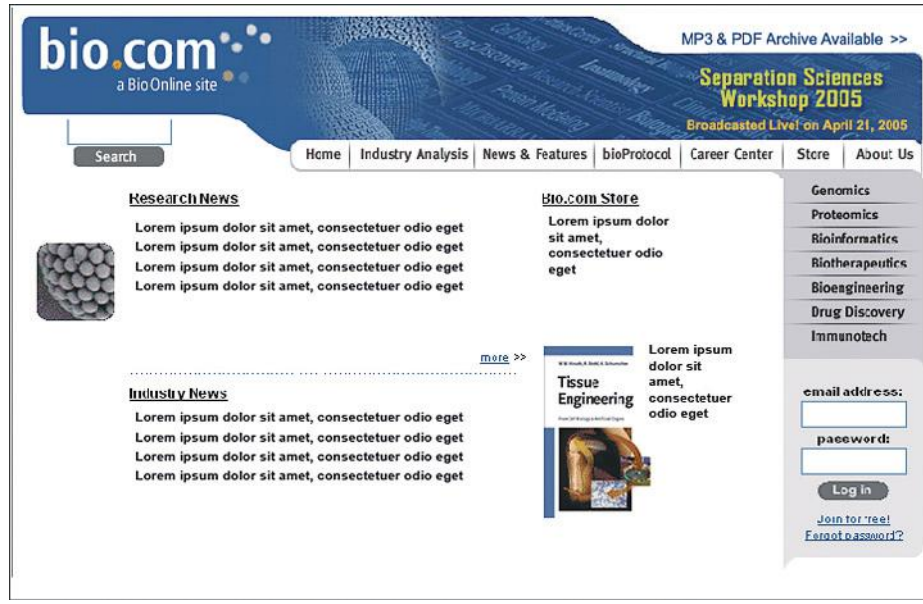
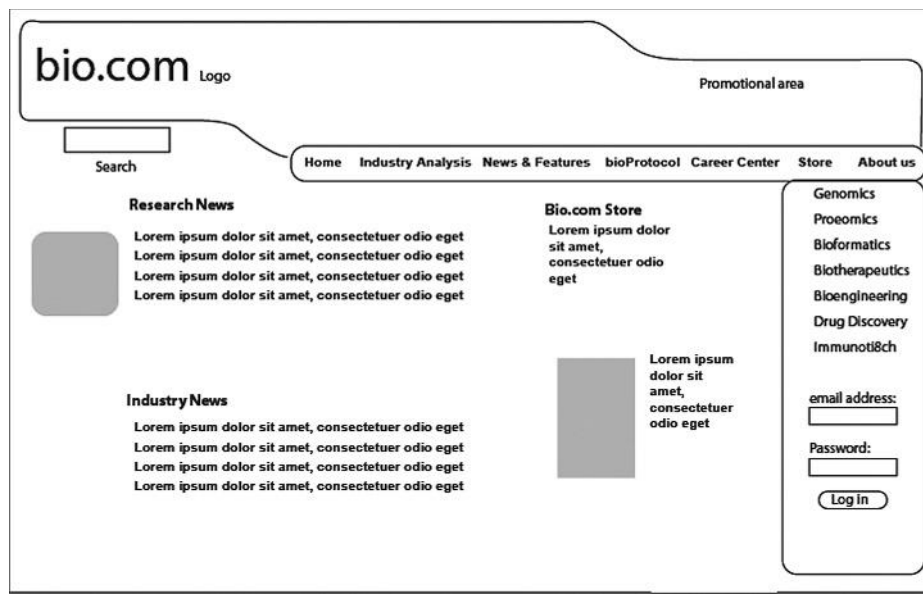


FIGURE 5.8 *Visual design deemphasized in a prototype.*



EDITORIAL CONTENT

Editorial content is the text that appears in the interface, e.g., labels, headers, titles, messages, and notifications. Editorial content also includes the usage of text, including style, flow, and expression.

Style is the type of content and the format it's presented in, such as an order form, a news story, an advertisement, etc. Style also indicates how information is syntactically presented to the user. Does it address the user as a professional or as a friend or as a salesman?

Flow consists of the method used to carry a reader through a body of content. If the content is a form with data entry fields, then flow is how the terminology guiding the user to complete the task of data entry or editing. Flow depends on the arrangement of fields and how well they are organized, grouped, labeled, and laid out. If the editorial content is in narrative or story format, flow depends on organization, presentation, and continuity in addition to any navigation required for a user to move to other screens that are part of the narrative.

Editorial expression is the voice of the content, e.g., technical, entertaining, legal, instructive, and so on.

Questions you want to ask when considering style are who is the audience and their background? What is the most important thing you need to communicate with style? Is it trust, friendliness, nostalgia, or other emotional attributes? What are the expectations for the tone of voice users would expect under a real-life situation? For example, what would the tone of a banker be and compare with the tone expressed via the bank website? What would a ceramics sales person say compared with the style expressed by a ceramics website?

Techniques to Adjust the Fidelity of the Editorial Content

Similar to other content components of a user interface, editorial content can be emphasized or deemphasized depending on the objectives of the prototype. The emphasis can range along spectrums of visual fidelity, editorial finish, and accuracy of message flow and layout. A prototype created later in the software design process typically has final or near-final edited text fully rendered in the selected font styles, weights, sizes, and colors (see Figure 5.3) to enable the usability test participant to fully experience the style, flow, and expression of text and language intended for the final product.

A prototype created early in the software process typically has the editorial content represented as shaded boxes or greeked text as shown in Figure 5.5. This more abstract representation of text is used to deemphasize the editorial content so that viewers focus on the emphasized aspects of content. This abstraction prevents viewers from reading unfinished editorial content, usually only partially conceived and written during the early conceptual phase of the software effort. In prototypes built to focus on interaction and navigation, for example, some parts of the editorial content, such as titles, subtitles, and labels, may be rendered in a higher fidelity, whereas blocks of running narrative text are represented abstractly. This allows a user to focus on navigation elements and cues on a screen and across screens while inhibiting the reading of text.

BRAND EXPRESSION

Branding is "an aggregation of all the physical and emotional characteristics of a company, a product, or a service encountered by the consumer at all points of contact. Branding is important because it communicates a brand's business proposition and, hopefully, a reason why a consumer should desire the product represented by the brand" [Roellig 2001]. The branding of products and services

is often misunderstood as merely slapping on a company's logo and color scheme. Branding is actually much more. To better understand branding, we share some definitions from the American Institute of Graphic Arts' (AIGA) book, *The Dictionary of Brand* [Neumeier 2004]:

- *Branding*: a person's perception of a product, service, experience, or organization; the art and science of brand building
- *Brand alignment*: the practice of linking brand strategy to customer touch points
- *Brand asset*: a distinctive feature of a product, service, company, or brand
- *Brand attribute*: a distinctive characteristic of a product, service, company, or brand
- *Brand experience*: all the interactions people have with a product, service, or organization; the reaction to a brand by the marketplace

Given the definitions above, we believe it is impossible to divorce branding from the other prototyping content presented in this chapter. By both the definition of brand and branding, software certainly qualifies as a major source of user exposure to an organization and its strategy, including the positive or negative associations with companies, products, and services. Interaction with software is sometimes the most intimate experience users have with a company. Therefore, how an organization expresses itself through its products and services are valuable and memorable experiences in terms of brand exposure. From this perspective, we can see why ensuring the highest quality in the user interface and user experience of software is as much a brand issue as it is a usability issue. This emphasizes the importance of including and evaluating branding as part of any planned iterative prototyping activities.

Increasing Brand Fidelity

Brand fidelity can be increased by taking an inventory of the branding elements of all the content areas and ensuring that they align and evoke desired reactions (see Figure 5.7). Examples of branding elements are:

- Logos
- Corporate colors
- Company slogans
- Company terminology
- Company standards and guidelines
- Corporate style guides

Decreasing Brand Fidelity

To decrease brand fidelity, ensure that branding elements are either excluded from or blocked out of prototypes (see Figure 5.8). Or you can imagine that a certain

place would be a spot where company branding would be used, so use a text callout in the prototype but do not actually portray it. You can use the content-specific fidelity techniques such as using a placeholder instead of the actual logo.

PRESENTING THE WRONG CONTENT IN A PROTOTYPE

A highly respected design consultant arrived at work on a project that involved rebranding a business-to-business website designed for petroleum engineers, Petroleng. The branding design consultant planned to mock up a visual concept for Petroleng. She teamed with an interaction designer who was already working on the project's interaction design.

The branding consultant used the interaction designer's new designs as the basis for her rebranding presentation because the interaction designer was working on the future of the product. Unfortunately, the interaction designer's work was quite controversial due to its use of experimental design concepts. Instead of greeking the interface content, the branding designer opted to include the complete content of the prototype so that the branding would be seen in context. The branding review meeting was prefaced by valiant attempts to set expectations.

"Just pay attention to the visual direction here and the visual branding elements."

These attempts were all in vain; the natural urge to read the text and evaluate the suggested interaction proved too powerful. The meeting degenerated from a high-level branding discussion of Petroleng to discussing the contentious interaction design and its details. This misguided focus on interaction design was more emphatic than usual due to the addition of the branding elements and the fear of finalized design decisions already existing. Instead of the prototype contents providing context to the branding discussion, the opposite happened. The branding gave context to the contentious interaction design, sparking a discussion on that instead of the branding. After the meeting the primary topic of branding was hardly discussed, although everyone congratulated the consultant on a great presentation of the interaction design. Clearly, a discussion and presentation on interaction design needed to take place, but it had occurred at the expense of the branding concept when it did not need to.

The branding design consultant was quite expensive, and there was little budget remaining for another round of work and an additional presentation. Furthermore, the company simply could not afford any more time in the schedule to revisit branding. Consequently, sign off on the whole design was forced to occur in less than ideal conditions. The consultant acknowledged that the client was happy, but she was unhappy due to the client's association of her with the great meeting on interaction design, not the rebranding effort she was hired for. Petroleng could have gotten much more for their branding money if the branding consultant had left the detailed information content out of her prototype.

SYSTEM PERFORMANCE/BEHAVIOR

System performance and behavior are characteristics of the users' experiences with the technical expression of software. How long does it take to complete a save interaction? Does a user error force the user to call tech support, or is a resolution suggested for the user?

System performance can be affected by a number of variables, including the quality of the programming code, throughput of the system, network bandwidth and traffic, and the data management and handling. System performance plays an important role in user perception of the software.

Slow performance and system response lead to dissatisfaction or, at least, vulnerability of your product to an aggressive competitor. Therefore, it makes no sense to skimp on investing in improving the system performance. Designers should take into consideration the technical requirements: web pages that behave like window–object interfaces usually have performance issues. A graphically rich website that is mostly accessed by dial-up connections will also have performance issues. Because system performance and behavior are such vitally important issues, they should be accounted for as part of the system requirements during the design.

Who owns these system issues? The interaction designer should own the system behavior issues. The wording of error messages or other system dialogs is usually the technical writer's domain. The systems performance issues belong to engineering because that group is in the best position to judge which system performance issues are affected by the design.

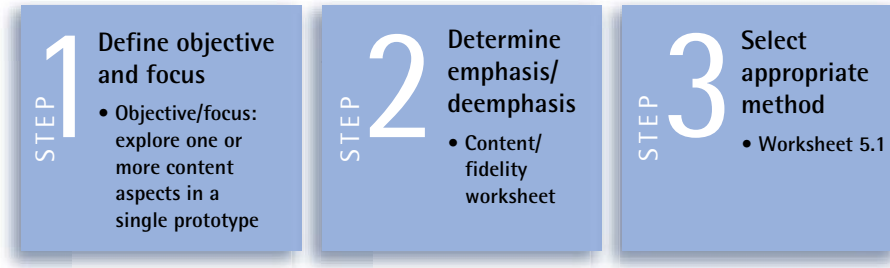
Techniques to Adjust the Fidelity of the System Performance/Behavior

In general, the only real method to test system performance is via an actual coded application. To provide a real sense of the system performance, this prototype (or beta release) should have all the system connections and actual data of the working application. Many people believe it is too late in the game to make huge system changes at this point because the code freeze or system test phases are near. Thus, the design team is obliged to ensure that engineering, the owners of the system performance, is involved in the early stages of design so they can raise any performance considerations and hopefully have time to resolve them without adversely affecting the design.

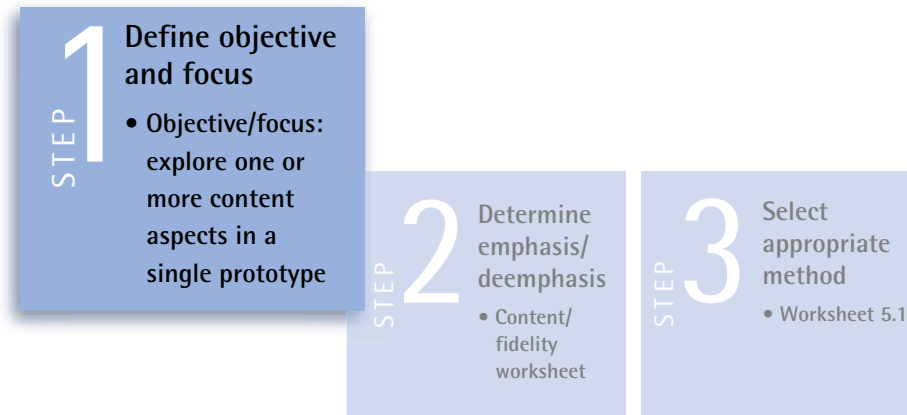
In the early stages of software design, a design team should not be unnecessarily encumbered by system performance and behavior emulation. Low-fidelity prototyping of just about any sort deemphasizes system performance and behavior. This lack of system emphasis enables the early most innovative stages of design to avoid being hampered by technical limitations, which may or may not be resolved in the course of developing the software. To free the design team from the confines of an actually working coded system, lower fidelity prototyping like storyboards, wireframes, and paper prototypes can be used.

HOW TO SELECT THE RIGHT PROTOTYPE CONTENT EXPRESSION

As you've just read, the content expression of a prototype can be tuned and modulated to emphasize or deemphasize various aspects and combinations of the content to receive feedback on a specific design goal. This tuning compels the viewer to concentrate on the emphasized aspects of a prototype while other aspects recede into the background.

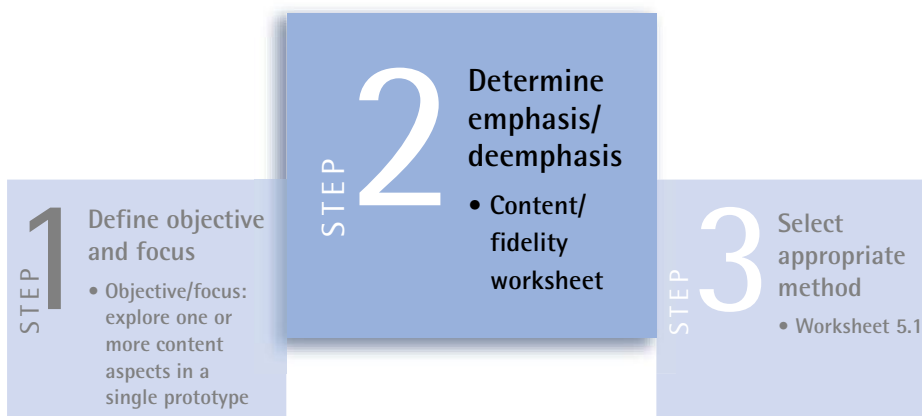


Step 1: Define objective and focus



Define the objective and focus of your prototype. For example, if you are designing a website, the objective could be tactical, such as focusing on navigation and task flow and page layout and organization. Or, the objective could be strategic with a focus on the branding scheme, the overall software conceptual model, and so on.

Step 2: Determine emphasis/deemphasis



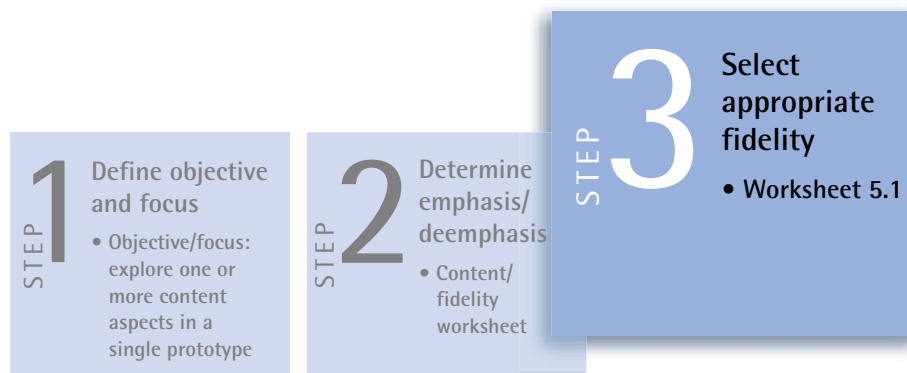
Based on your objective and focus, determine what aspects of the prototype content you want to emphasize and deemphasize. For example, you could choose to explore mapping interaction and navigation to a task flow and focus on interaction design and navigation within a web page as well as among pages. To have your prototype reflect your objectives, you want to emphasize the following content components:

- Interaction design and navigation model
- Information design

In turn, to avoid undue distraction in this example scenario of exploring interaction and navigation, you want to deemphasize:

- Editorial content
- Visual design
- Branding
- System performance and behavior

Step 3: Select appropriate fidelity



Now that you've determined what to emphasize and deemphasize in the prototype, you need to select an appropriate prototyping fidelity that allows you to achieve these objectives. You want to choose a fidelity that allows you to tune the content components to achieve your desired focus.

Worksheet 5.1 will help you select the best fidelity. Enter the information in this worksheet following steps 1–6 in the left hand column. In filling out the Content/Fidelity matrix at the top, entitled *Planned*, indicate what your audience will need in terms of fidelity for each content item. In the *Actual, current state* matrix, you will try to capture the needs given the audience needs and what you will be able to accomplish. You may wish to keep this matrix blank until the next Chapter, when going over the rest of the prototyping characteristics may alter the fidelity you can produce.

WORKSHEET 5.1: Determining Content fidelity

Step			
1	**Project Name:**		**Author:**
1	**Project Date:**		**Internal Ref.:**
1	**Product Name:**		**Target Release:**
1	**Current Phase:**		
2	**Target Audience**	**Internal** ☐ Design team members ☐ Upper Management ☐ Lead designers ☐ Product Managers ☐ Marketing & Sales ☐ Developers ☐ Technical writers ☐ Domain specialists/Analysts ☐ QA engineers ☐ Business/financial stakeholders	**External** ☐ End-user (consumers) ☐ Customers (purchasers) ☐ Domain specialists/ Analysts ☐ Financial stakeholders ☐ Business analysts ☐ Press
1	**Overall prototype objective:**	The purpose of this prototype is to show XXX to YYY in order to ZZZ.	
2	**Audience needs**	This audience best relates to prototypes that XXX	
2	**Audience aversions**	This audience is least likely to need a prototype that XXX	

Audience importance

Content		Not needed	Very low	Low	Medium	High	Very High (essential)			
Planned										
3	Information									
4	Interaction									
5	Visual Design									
3	Editorial									
5	Branding									
4	System									
		Not needed	**Very low**	**Low**	**Medium**	**High**	**Very High**			
6 Actual, current state	Information									
6	Interaction									
6	Visual Design									
6	Editorial									
6	Branding									
6	System									

SUMMARY

In Step 3 of the effective prototyping process, we discussed the prototyping content considerations you need to address to best achieve your prototyping needs. This discussion included mixing fidelities and other variables to emphasize and deemphasize different aspects of a prototype for usability and presentation purposes. The different types of content covered in the chapter were:

- Information design
- Interaction design and navigation model
- Visual design
- Editorial content
- Branding
- System performance/behavior

ARNOSOFT TEAM TACKLES CONTENT

Reed, Ina, and Dirk are discussing the next prototype iteration. The storyboard was a success, but Reed still wants something more "clickable." To understand exactly what Reed wants, Dirk and Ina review the prototyping content and the fidelity level that Reed requires to get the prototype he wants and that the team needs.

First, Ina and Dirk discuss the target prototype audience with Reed. Reed at first just says, "Myself."

Dirk in disbelief says, "You want us to do this just for you?"

Ina adds, "I think other people might be interested in seeing this as well."

So together they look at the list of potential stakeholders for this prototype. They cross out the ones they believe are not going to be relevant for this prototype:

Internal

- □ ~~Design team members~~
- □ Upper Management
- □ ~~Lead designers~~
- □ ~~Product Managers~~
- □ ~~Marketing & Sales~~
- □ Developers
  - □ ~~Technical writers~~
  - □ Domain specialists/Analysts
  - □ ~~QA engineers~~
  - □ Business/financial stakeholders

External

- □ End-user (consumers)
- □ Customers (purchasers)
- □ ~~Domain specialists/Analysts~~
- □ ~~Financial stakeholders~~
- □ ~~Business analysts~~
- □ ~~Press~~

Reed explains he wants to make sure this site can "fly" so he is actually quite eager to also see how end users will use this prototype. In the review of information content, Reed points out that the information needed is very important. Dirk is also very concerned about the real needs

of the user regarding their keyword usage. So the top priority for audience is the defined user and the internal domain specialist (a.k.a. Reed). Therefore the information fidelity that the team considered was as follows:

Information: High
Editorial: Medium

The team next looked at the interaction content from their desire to have the interaction design be high, but because they ultimately did not want to sweat implementation details (or use precious development resources), they decided that system performance could be low fidelity. After that, they considered the visual content and agreed that the visual design and branding could be left until later, especially while Reed and Vista argued whether it should be called Dish's Dishes or CeramaSite. Finally, they did a gap analysis of the required content fidelity with what they currently possessed (see completed Worksheet 5.1 below).

After this exercise, Dirk was sure they needed an HTML prototype. Reed wanted them to just go ahead and build it. Finally, Ina was convinced a series of wireframes shown to a focus group would probably suffice. Here is what the finished content fidelity worksheet for ArnoSoft looks like:

REFERENCES

Donis A. Dondis. A Primer of Visual Literacy. Cambridge, MA: MIT Press, 1973.

Armin Hofmann. Graphic Design Manual: Principles and Practice. New York: Van Nostrand Reinhold, 1965.

Marty Neumeier. The Dictionary of Brand. New York: AIGA Center for Brand Experience, 2004, pp. 15–17.

Larry Roellig. Designing global brands: critical lessons. Design Management Journal, 2001, Fall.

Louis Rosenfeld, Peter Morville. Information Architecture for the World Wide Web: Designing Large-Scale Web Sites (2nd ed.). Sebastopol, CA: O'Reilly, 2002.

William Strunk, Jr., E.B. White. The Elements of Style (4th ed.). New York: Longman, 2000.

PHASE

The question of what to prototype relates to classic problem reduction. The designer must reduce the problem space into an ordered set of manageable subproblems.

–Bill Buxton