

Not only can this use of metaphor make for a better immediate discussion, but people are more likely to talk about the ideas with colleagues after the presentation itself, spreading the concepts like a virus. For example, throughout this chapter we have been referring to strategy as a “bridge”—this is obviously a metaphor, meant to make abstract concepts more concrete and memorable. You, too, can use metaphors to make your strategy easier to discuss.

Now, finally, you can congratulate the visionary within you, take a brief rest, and prepare for the detail orientation of the design and documentation phase.

## Recap

Let’s recap what we’ve learned in this chapter:

- An information architecture strategy serves as a bridge between research and design.
- The IA strategy provides a high-level conceptual framework for structuring and organizing an information environment.
- You should start considering possible strategies for structuring and organizing the product before research begins.
- The main deliverable of the strategy phase is the strategy report.
- We find it useful to create a project plan for the design of the information architecture as part of the strategy phase.
- You’re not done when you’ve created the report—you also need to present and discuss it with stakeholders.

## CHAPTER 13

# Design and Documentation

*You can use an eraser on the drafting table  
or a sledge hammer on the construction site.*

—Frank Lloyd Wright

In this chapter, we’ll cover:

- The role of diagrams in the design phase
- Why, when, and how to develop sitemaps and wireframes, the two most common types of IA diagrams
- How to map and inventory your content
- Content models and controlled vocabularies for connecting and managing granular content
- Ways to enhance your collaboration with other members of the design team
- Style guides for capturing your past decisions and guiding your future ones

When you cross the bridge from research and strategy into design, the landscape shifts dramatically. The emphasis moves from process to deliverables as your clients and colleagues expect you to move from thinking and talking to actually producing a clear, well-defined information architecture.

This can be an uneasy transition. You must relinquish the white lab coat of the researcher, leave behind the ivory tower of the strategist, and delve into the exposed territory of creativity and design. As you

commit your ideas to paper, it can be scary to realize there's no going back. You are now actively shaping what will become the user experience. Your fears and discomforts will be diminished if you've had the time and resources to do the research and develop a strategy; if you're pushed straight into design (as is too often the case), you'll be entering the uncertain realm of intuition and gut instinct.

It's difficult to write about design because the work in this phase is so strongly defined by context and influenced by tacit knowledge. You may be working closely with a graphic designer to create a small website or app from the ground up. Or you may be building a controlled vocabulary and index as part of an enterprise-level redesign of a broad information environment that involves more than a hundred people. The design decisions you make and the deliverables you produce will be informed by the total sum of your experience.

In short, we're talking about the creative process. Ours is a vast, complex, and ever-changing canvas. Often, the best way to teach art is through the time-tested practice of show-and-tell. So, in this chapter, we'll use work products and deliverables to tell the story about what we do during the design phase.

Before we dive in, here's a caveat: although this chapter focuses on deliverables, process is as important during design as it is during research and strategy. This means that the techniques covered previously should be applied to these later phases, albeit with more concrete and detailed artifacts—ranging from vocabularies to wireframes to working prototypes—being tested.

And another caveat: for reasons beyond your control, you'll occasionally—even frequently—find yourself in the uncomfortable situation of bypassing research and strategy altogether, skipping headlong into the abyss of design. Deliverables are especially critical in this context; they're anchors that, by forcing the team to pause, capture, and review its work, regulate and moderate an out-of-control project. You can also use deliverables to unmask design problems and force the project to backtrack to research and design tasks that should have been handled much earlier.

## Guidelines for Diagramming an Information Architecture

We are under extreme pressure to clearly represent the product of our work. Whether it's to help sell the value of information architecture to a potential client or to explain a design to a colleague, we rely upon visual representations to communicate what it is we actually do.

And yet information architectures—as we've mentioned many times—are abstract, conceptual things. Websites, in particular, are not finite; often you can't tell where one ends and the other begins. Subsites and the “invisible web” of databases further muddy the picture of what should and shouldn't be included in a specific architecture. Digital information itself can be organized and repurposed in an almost infinite number of ways, meaning that an architecture is typically multidimensional—and therefore exceedingly difficult to represent in a two-dimensional space such as a whiteboard or a sheet of paper.

So we're left with a nasty paradox: we're forced to demonstrate the value and essence of our work in a visual medium, though our work itself isn't especially visual.

There really is no ideal solution. The field of information architecture is too young and dynamic for its practitioners to have figured out how best to visually represent information architectures, much less agree upon a standard set of diagrams that work for all audiences in all situations.<sup>1</sup> And it's unlikely that the messages we wish to communicate will ever lend themselves easily to 8.5" × 11" sheets of paper.

Still, there are a couple of good guidelines to follow as you document your architecture:

---

<sup>1</sup> For an in-depth look at deliverables, we recommend Dan Brown's *Communicating Design: Developing Web Site Documentation for Design and Planning, Second Edition* (San Francisco: New Riders, 2010). Dan is an information architect whose work is highly respected by many practitioners.

### *Provide multiple “views” of an information architecture*

Information environments are too complex to show all at once; a diagram that tries to be all things to all people is destined to fail. Instead, consider using a variety of techniques to display different aspects of the architecture. No single view takes in the whole picture, but the combination of multiple diagrams might come close.

### *Develop those views for specific audiences and needs*

You might find that a visually stunning diagram is compelling to client prospects, therefore justifying its expense. However, it probably requires too many resources to use in a production environment, where diagrams may change multiple times per day. Whenever possible, determine what others need from your diagrams before creating them. For example, Keith Instone, an information architect formerly at IBM, created very different diagrams for communicating “upstream” with stakeholders and executives than for communicating “downstream” with designers and developers.

Whenever possible, present information architecture diagrams in person, especially when the audience is unfamiliar with them. If you can’t be there in person, at least be there via videoconference or telephone. Again and again, we’ve witnessed (and suffered from) huge disconnects between what the diagram was intended to communicate and what it was actually understood to mean. This shouldn’t be surprising, because, as we mentioned, there is no standard visual language to describe information architectures yet. So, be present to translate, explain, and (if necessary) defend your work.

Better yet, work in advance with whomever you’re presenting your diagrams to—clients, managers, designers, programmers—so they can understand what they will need from them. You may find that your assumptions of how they would use your diagrams were quite wrong. We’ve seen a large, respected firm fired from a huge project because it took too many weeks to produce bound, color-printed, sexy diagrams. The client preferred (and requested) simple, even hand-drawn, sketches because it needed them as soon as possible.

As we’ve seen in previous chapters, the most frequently used diagrams are sitemaps and wireframes, which focus more on the structure of content than its semantic value. Because of this, sitemaps and wireframes are not as effective at conveying the semantic nature of

content or labels. We’ll discuss both types of diagrams in detail in the following sections, but first it’s helpful to understand the visual language that these diagrams use.

## Communicating Visually

Diagrams are useful for communicating the two basic aspects of an information system’s structural elements.<sup>2</sup> Diagrams define:

### *Content components*

What constitutes a unit of content, and how those components should be grouped and sequenced

### *Connections between content components*

How content components are linked to enable actions such as navigating between them

No matter how complex your diagrams may ultimately become, their main goal will always be to communicate what your information environment’s content components are and how they’re connected.

A variety of visual vocabularies have emerged to help convey the complexity of information architecture in visual diagrams, each providing a clear set of terms and syntax to visually communicate components and their links. The best known and most influential visual vocabulary is Jesse James Garrett’s (<http://www.jjg.net/ia/visvocab>), which has been translated into eight languages. Jesse’s vocabulary anticipates and accommodates many uses, but perhaps the greatest reason for its success is its simplicity; just about anyone can use it to create diagrams, even by hand.

Visual vocabularies are at the heart of the many templates used to develop sitemaps and wireframes. Thanks to their developers’ generosity, there are many free templates you can use to create your own deliverables; Table 13-1 provides useful examples. Each requires one of the common charting programs, like Microsoft’s Visio (for Windows PCs) or Omni Group’s OmniGraffle (for Macs).

<sup>2</sup> Semantic aspects, like controlled vocabularies, don’t lend themselves as easily to visual representation.

Table 13-1. Templates for common diagramming tools

| Name                              | Creator   | Application | URL   |
|-----------------------------------|---|-------------|---|
| OmniGraffle<br>Wireframe Stencils | Michael Angeles                                   | OmniGraffle | <a href="http://bit.ly/omnigraffle_wireframe">http://bit.ly/omnigraffle_wireframe</a>       |
| Sitemap Stencil                   | Nick Finck  | Visio       | <a href="http://www.nickfinck.com/stencils.html">http://www.nickfinck.com/stencils.html</a> |
| Wireframe Stencil                 | Nick Finck  | Visio       | <a href="http://www.nickfinck.com/stencils.html">http://www.nickfinck.com/stencils.html</a> |
| Block Diagram<br>Shapes Stencil   | Matt Leacock, Bryce<br>Glass, and Rich<br>Fulcher | OmniGraffle | <a href="http://www.paperplane.net/omnigraffle/">http://www.paperplane.net/omnigraffle/</a> |
| Flow Map Shapes<br>Stencil        | Matt Leacock, Bryce<br>Glass, and Rich<br>Fulcher | OmniGraffle | <a href="http://www.paperplane.net/omnigraffle/">http://www.paperplane.net/omnigraffle/</a> |

What if you're a nonvisual person who cringes at the idea of learning OmniGraffle? Or the people you're communicating your ideas to aren't visually oriented? Does your work *have* to be visual?

Absolutely not. As ugly as the results may be, you can render your sitemaps as outlines in a word processor or use a spreadsheet's cells in a similar fashion. You can write page descriptions that cover the same bases as your wireframes—just about anything can be rendered in text. Ultimately, these deliverables are first and foremost communication tools. You need to play to your own communication strengths and, more importantly, take advantage of whatever style works best for your audience.

But remember, there's a reason they say "a picture is worth a thousand words." The lines between information architecture and the more visual aspects of design are blurry, and at some point, you'll have to connect your IA concepts, however textual, to the work that is the responsibility of graphic designers and interaction designers. Hence, we spend most of our time in this chapter on visual means for communicating information architectures.

Sitemaps

Sitemaps show the relationships between information elements such as pages and other content components, and can be used to portray organization, navigation, and labeling systems. Both the diagram and the navigation system display the "shape" of the information

space in overview, functioning as a condensed map for site developers and users, respectively.

High-Level Architecture Sitemaps

High-level sitemaps are often created as part of a top-down information architecture process (and they may also be produced during a project's strategy phase.) Starting with the main page, you might use the process of developing a sitemap to iteratively flesh out more and more of the architecture, adding subsidiary sections, increasing levels of detail, and working out the navigation from the top down. Sitemaps can also support bottom-up design, such as displaying a content model's content chunks and relationships; we discuss these uses later in the chapter.

The very act of shaping ideas into the more formal structure of a sitemap forces you to be realistic and practical. If brainstorming takes you to the top of the mountain, creating the sitemap can bring you back down to the valley of reality. Ideas that seemed brilliant on the whiteboard may not pan out when you attempt to organize them in a practical manner. It's easy to throw around concepts such as "personalization" and "adaptive information architectures." It's not so easy to define on paper exactly how these concepts will be applied to a specific product.

During the design phase, high-level sitemaps are most useful for exploring primary organization schemes and approaches. High-level sitemaps map out the organization and labeling of major areas, usually beginning with a bird's-eye view from the main page of the website. This exploration may involve several iterations as you further define the information architecture.

High-level sitemaps (like the one in Figure 13-1) are great for stimulating discussions focused on the organization and management of content as well as on the desired access pathways for users. These sitemaps can be drawn by hand, but we prefer to use diagramming software such as Visio or OmniGraffle. These tools not only help you to quickly lay out your architecture sitemaps, but can also help with site implementation and administration. They also lend your work a more professional look, which, sadly, can sometimes be more important than the quality of your actual design.

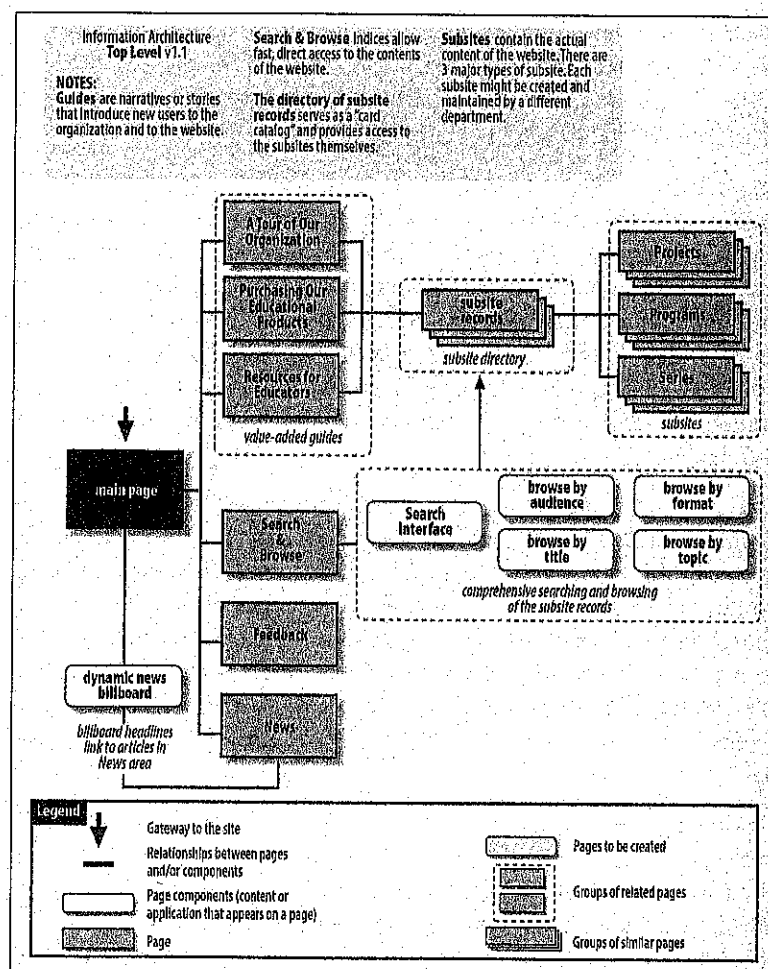


Figure 13-1. A high-level sitemap

Let's walk through the sitemap in Figure 13-1 as if we were presenting it to clients or colleagues. The building block of this architecture is the subsite. Within this company, the ownership and management of content is distributed among many individuals in different departments. There are already dozens of small and large websites, each with its own graphic identity and information architecture. Rather than trying to enforce one standard across this collection of sites, this sitemap suggests an "umbrella architecture" approach that allows for the existence of lots of heterogeneous subsites.

Moving up from the subsites, we see a directory of subsite records. This directory serves as a "card catalog" that provides easy access to the subsites. There is a record for each subsite; each record consists of fields such as *title*, *description*, *keywords*, *audience*, *format*, and *topic*, which describe the contents of that subsite.

By creating a standardized record for each subsite, we are actually creating a database of subsite records. This database approach enables both powerful known-item searching and exploratory browsing. As you can see from the Search & Browse page, users can search and browse by title, audience, format, and topic.

The sitemap also shows three guides. These guides take the form of simple narratives or "stories" that introduce new users to the site's sponsor and selected areas within the website.

Finally, we see a dynamic news billboard that rotates the display of featured news headlines and announcements. In addition to bringing some action to the main page, this billboard provides yet another way to access important content that might otherwise be buried within a subsite.

At this point in the discussion of the high-level sitemap, you are sure to face some questions. As you can see, sitemaps don't completely speak for themselves, and that's exactly what you want. High-level sitemaps are an excellent tool for explaining your architectural approaches and making sure that they're challenged by your client or manager. Questions such as "Do those guides really make sense, considering the company's new plans to target customers by region?" give you an excellent opportunity to gain buy-in from the client and to fireproof your design from similar questions that might arise much later in the process, when it'll be more expensive to make changes.

Presenting sitemaps in person allows you to immediately answer questions and address concerns, as well as to explore new ideas while they're still fresh. You might also consider augmenting your sitemaps with a brief text document to explain your thinking and answer the most likely questions right on the spot. At the very least, consider providing a "Notes" area (as we do in this example) to briefly explain basic concepts.



## Digging Deeper into Sitemaps

As you create sitemaps, it's important to avoid getting locked into a particular type of layout. Instead, let form follow function. Notice the difference between Figures 13-2 and 13-3.

Figure 13-2 provides a holistic view of the information architecture for a global consulting firm. It's part of an initiative to build support for the overall vision of unified access to member firms' content and services. In contrast, Figure 13-3 focuses on a single aspect of navigation for The Weather Channel's website, aiming to show how users will be able to move between local and national weather reports and news. Both sitemaps are high level and conceptual in nature, yet each takes on a unique form to suit its purpose.

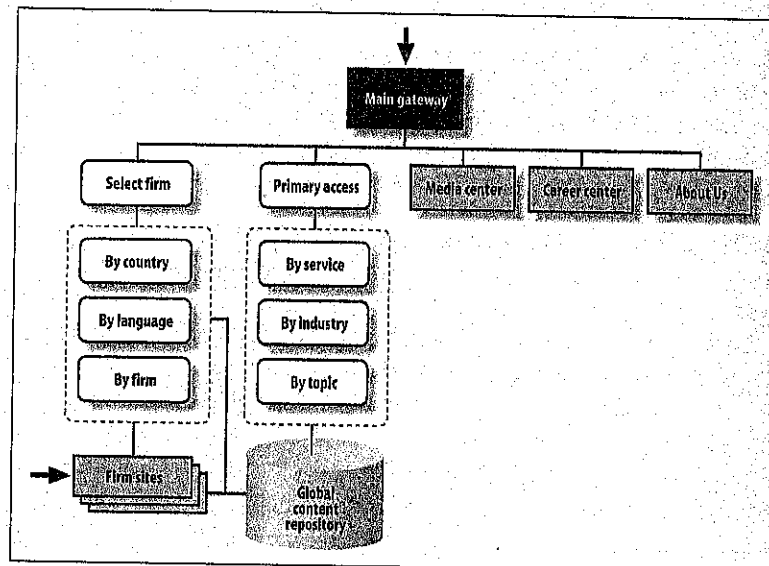


Figure 13-2. This sitemap illustrates the big picture for a consulting firm's public site...

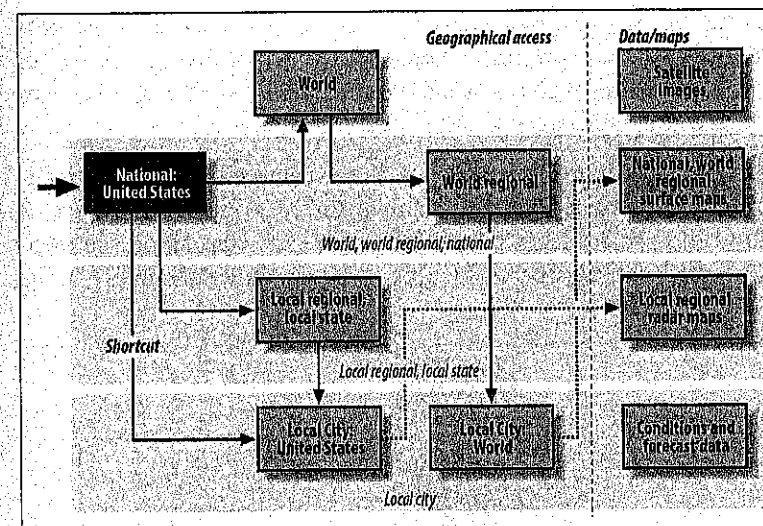


Figure 13-3. ...while this one focuses on geographic hub navigation for The Weather Channel's site...

In Figure 13-4, we see a high-level sitemap for the online greeting card website Egreetings.com. This sitemap focuses on the user's ability to filter cards based on format or tone at any level while navigating the primary taxonomy.

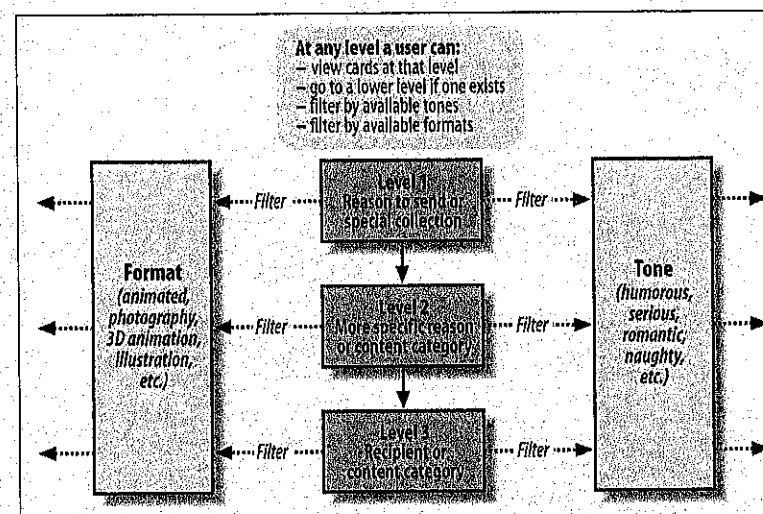


Figure 13-4. ...and this one demonstrates how filtering might work at Egreetings.com

It's important to remind ourselves that information environments aren't just about content; we can also contribute to the design of transactional and task-centered systems. This work requires task-oriented sitemaps and process maps.

For example, Figure 13-5 presents a user-centered view of the card-sending process at Egreetings.com prior to a redesign project. It allows the project team to walk through each step along the web- and email-enabled process, looking for opportunities to improve the user experience.

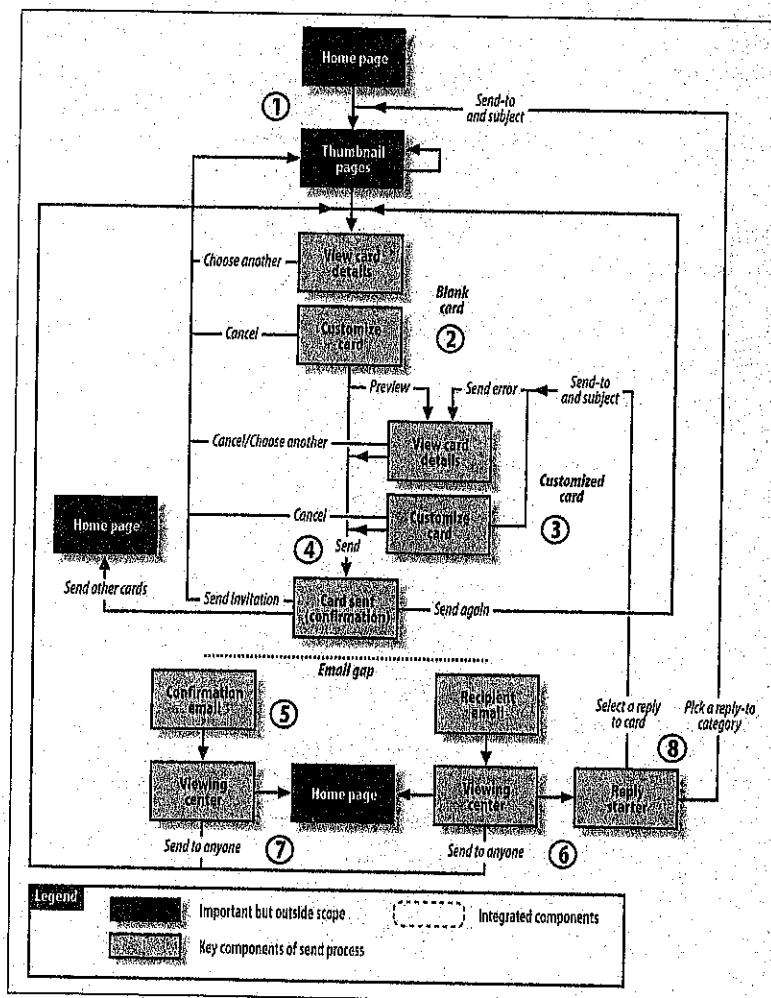


Figure 13-5. A task-oriented sitemap of the card-sending process

Figure 13-6 demonstrates how casual browsers may become engaged in a political campaign over time by interacting with its website's content. This sitemap is as much about changes in the user's mind as it is descriptive of the site's content and navigation.

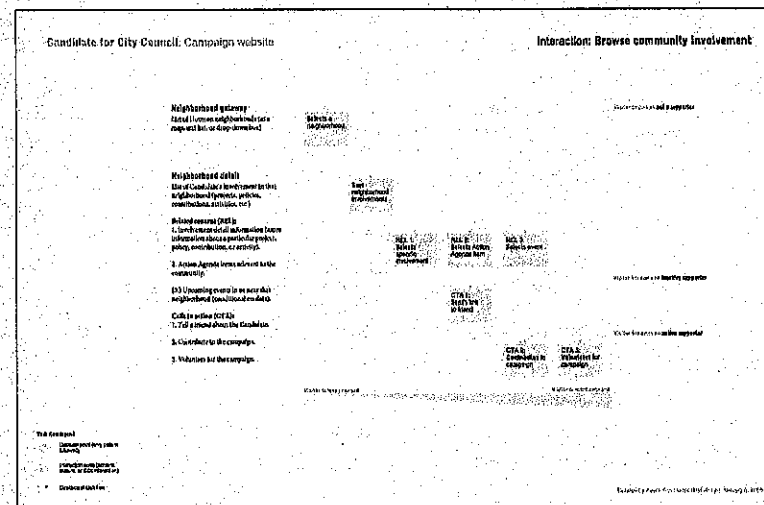


Figure 13-6. A sitemap by Austin Govella depicting growing levels of engagement in a political candidate's campaign

You'll notice that as we dug deeper, we moved from high-level site-maps toward diagrams that isolated specific aspects of the architecture, rather than communicating the overall direction of the site. Sitemaps are incredibly flexible; while boxes and connectors can't communicate everything about a design, they are simple enough that just about anyone can both develop and understand them.

You should also note that all of these sitemaps leave out quite a bit of information. They focus on the major areas and structures of the site, ignoring many navigation elements and page-level details. These omissions are by design, not by accident. Remember the rule of thumb for sitemaps: less is more.

## Keeping Sitemaps Simple

As a project moves from strategy to design to implementation, site-maps become more utilitarian. At this stage, they are focused more on communicating the information architecture to others involved in design and development, and less on strategy and product.

definition. “Lower-level” sitemaps need to be produced and modified quickly and iteratively, and often draw input from an increasing number of perspectives, ranging from visual designers to editors to programmers. Those team members need to be able to understand the architecture, so it’s important to develop a simple, condensed vocabulary of objects that can be explained in a brief legend. See Figure 13-7 for an example.

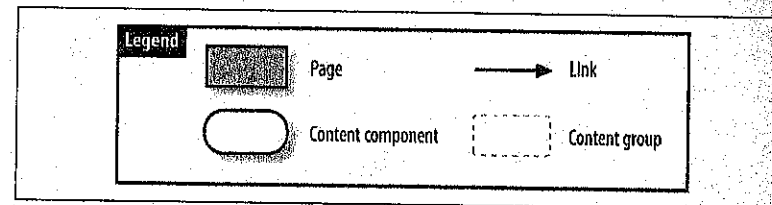


Figure 13-7. This sitemap legend describes an intentionally simple vocabulary

In this figure, the legend describes three levels of content granularity. The coarsest are content groups (made up of pages); these are followed by the pages themselves. Content components are the finest-grained content that it makes sense to represent in a sitemap. The arrow describes a link between content objects; these can be one-way or bidirectional links.

This is a minimal set of objects; we’ve found that retaining a limited vocabulary helps us avoid the temptation of overloading the diagram with too much information. After all, other diagrams can be used to convey other views of the architecture more effectively.

## Detailed Sitemaps

As you move deeper into the implementation stage, your focus naturally shifts from external to internal. Rather than communicating high-level architectural concepts to the client, your job is now to communicate detailed organization, labeling, and navigation decisions to your colleagues on the development team. In the world of “physical” architecture, this shift can be likened to architecture versus construction. You may work closely with the client to make big-picture decisions about the layout of rooms and the location of windows; however, decisions regarding the size of nails or the routing of the plumbing typically do not involve the client. And in fact, such minutiae often need not involve the architect either.

As with physical architecture, these small details often change on the construction site: perhaps the client has changed her mind about the size of her home office, or an electrical fixture is inconveniently located in the kitchen and must be moved. In any case, change is to be expected when abstract diagrams meet the real conditions of the construction site. In our field, agile and lean development methods call for rapid iteration, often based on incomplete information. Detailed sitemaps can (and should) evolve along with the rest of the design to address the new conditions and requirements that come up during the development process in these types of projects.

That said, you should try to map out the entire environment so that the production team can implement your plans as closely as possible when starting the development process. These sitemaps must present the complete information hierarchy from the main page to the destination pages. They must also detail the labeling and navigation systems to be implemented in each area of the environment.

Sitemaps will vary from project to project, depending upon the scope. On smaller projects, the primary audience for your sitemaps may be one or two graphic designers responsible for integrating the architecture, design, and content. On larger projects, the primary audience may be a technical team responsible for integrating the architecture, design, and content through a database-driven process. Let’s consider a few examples to see what sitemaps communicate and how they might vary.

Figure 13-8 shows a sitemap from the SIGGRAPH 96 Conference that introduces several concepts. By assigning a unique identification number (e.g., 2.2.5.1) to each component (e.g., pages and content chunks), the diagram presents the groundwork for an organized production process, ideally involving a database system that populates the website structure with content.

There is a distinction between a local and a remote page in Figure 13-8. A local page is a child of the main page on that sitemap, and inherits characteristics such as graphic identity and navigation elements from its parent. In this example, the Papers Committee page inherits its color scheme and navigation system from the Papers main page. On the other hand, a remote page belongs to another branch of the information hierarchy. The Session Room Layout page has a graphic identity and navigation system that are unique to the Maps area of the website.



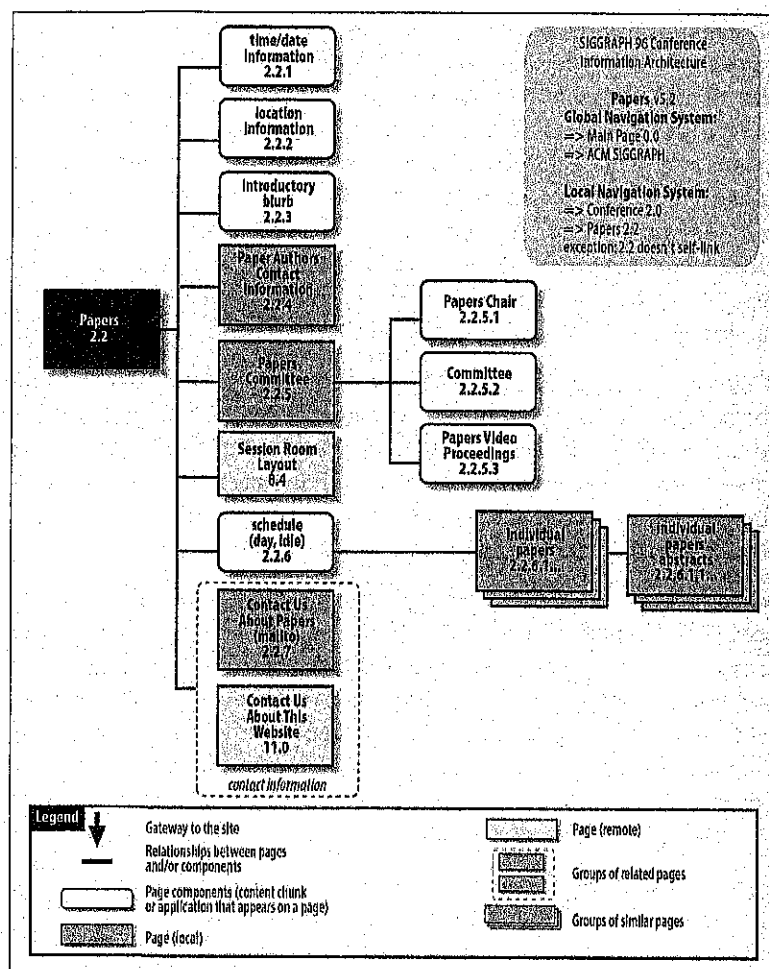


Figure 13-8. A sitemap of a major section of the SIGGRAPH conference website

Another important concept is that of content components or chunks. To meet the needs of the production process, it is often necessary to separate the content (i.e., chunks) from the container (i.e., pages). Content chunks such as “Contact Us About Papers” and “Contact Us About This Website” are sections of content composed of one or more paragraphs that can stand alone as independent packages of information. (We’ll discuss content chunking in more detail later in this chapter.) The rectangle that surrounds these content chunks indicates that they are closely related. By taking this

approach, the architect provides the designer with flexibility in defining the layout. Depending upon the space each content chunk requires, the designer may choose to present all of these chunks on one page, or create a closely knit collection of pages.

You may also decide to communicate the navigation system using these detailed sitemaps. In some cases, arrows can be used to show navigation, but these can be confusing and are easily missed by the production staff. A sidebar is often the best way of communicating both global and local navigation systems, as shown in Figure 13-8. The sidebar in the upper right of this sitemap explains how the global and local navigation systems apply to this area of the website.

## Organizing Your Sitemaps

As the architecture is developed, it needs to accommodate more than top-level pages. The same simple notation can be used, but how can you squeeze all of these documents onto one sheet of paper? Many applications will allow you to print on multiple sheets, but you’ll find yourself spending more time taping sheets together than designing. And if a diagram is too large to print on a single sheet, it’s probably also too large to reasonably view and edit on a standard monitor.

In this case, we suggest *modularizing* the sitemap. The top-level sitemap links to subsidiary sitemaps, and so on, and so on. These diagrams are tied together through a scheme of unique IDs. For example, in the top-level diagram in Figure 13-9, major pages are numbered x.0. For instance, the one representing “Committees and officers” is numbered 4.0. That page becomes the “lead page” on a new diagram (Figure 13-10), where it is also numbered 4.0. Its subsidiary pages and content components use codes starting with 4.0 in order to link them with their parent.

Using a unique identification scheme to tie together multiple diagrams helps us to somewhat mitigate the tyranny of the 8.5” × 11” sheet of paper, although you may still find that your architecture requires dozens of individual sheets of paper. This scheme can also be helpful for bridging a content inventory to the architectural process—content components can share the same IDs in both content inventory and sitemap. This means that in the production phase, adding content is not much different from painting by numbers.

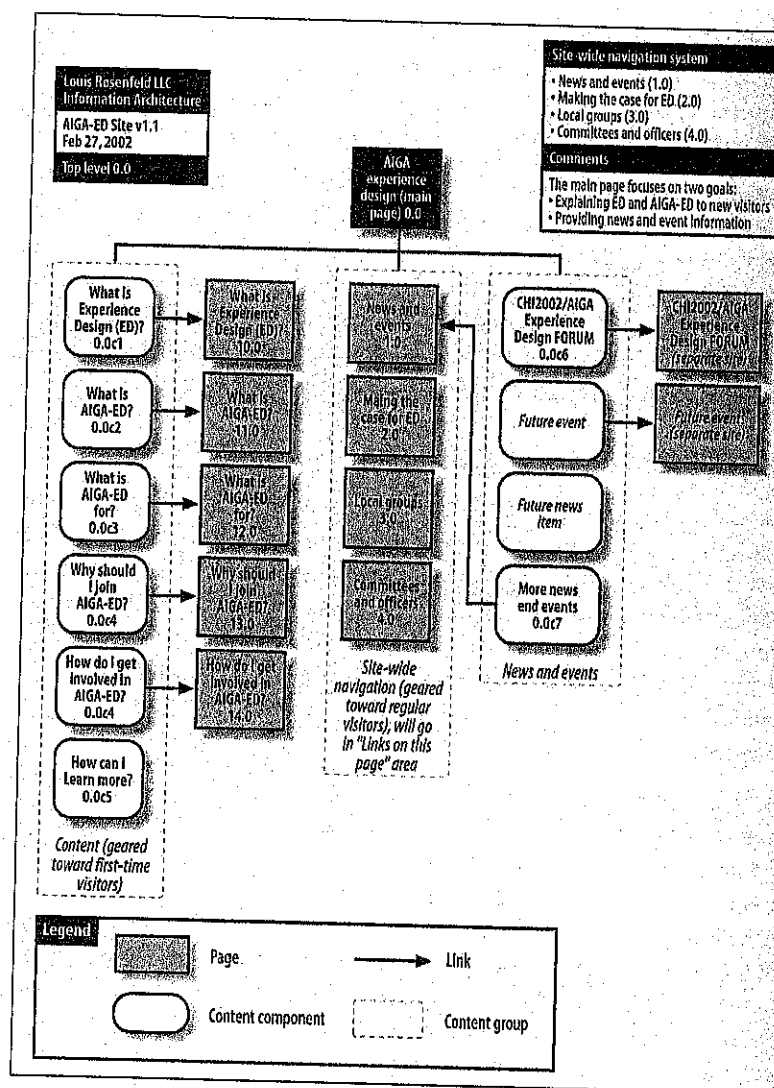


Figure 13-9. A detailed sitemap illustrating several concepts

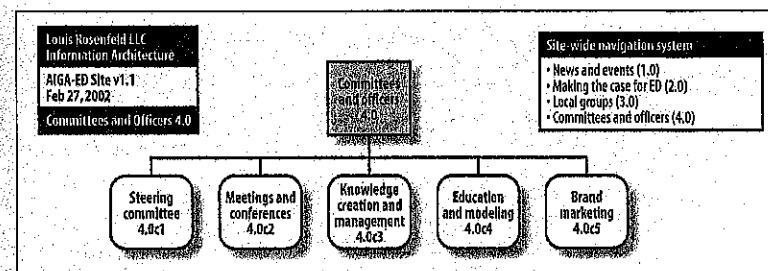


Figure 13-10. This subsidiary sitemap continues from the top-level sitemap

## Wireframes

Sitemaps can help you determine where content should go and how it should be navigated within the context of a website, subsite, app, or collection of content. Wireframes serve a different role: they depict how an individual page or template should look from an architectural perspective. Wireframes connect the product's information architecture with its interaction design.

For example, the wireframe forces you to consider such issues as where the navigation systems might be located on the page or screen. And now that you see it on an early layout, does it seem that there are actually too many ways to navigate? Trying out ideas in the context of a wireframe might force you back to the sitemap's drawing board, but it's better to make such changes on paper rather than reengineering the entire system at some point in the future.

Wireframes describe the content and information architecture to be included on the relatively confined two-dimensional spaces (e.g., pages, screens). Therefore, wireframes themselves must be constrained in size. These constraints force us to make choices about what components of the architecture should be visible and accessible to users; after all, if the architectural components absorb too much screen real estate, no room will be left for actual content!

Developing wireframes also helps clarify the grouping of content components, their order, and group priority. In Figure 13-11, "Reasons to Send" is of a higher priority than the "Search Assistant." This priority is made clear by the content's prominent positioning and the use of a larger typeface for its heading.