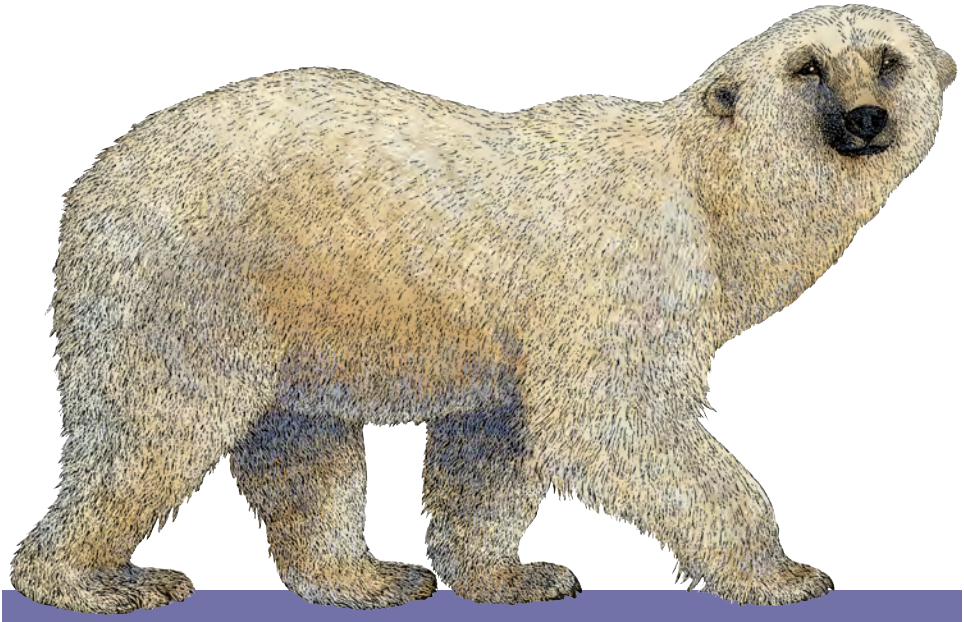
O'REILLY®

4th Edition



Information Architecture

FOR THE WEB AND BEYOND

Louis Rosenfeld,
Peter Morville & Jorge Arango

PART I

Introducing Information Architecture

Information is more abundant today than ever before. With smartphones, activity monitors, smart watches, tablets, and new Internet-enabled appliances of every kind, we also have many more ways of interacting with it than before. This abundance and pervasiveness makes our lives better in many ways, but it also introduces new challenges. With so much information available in so many places, it can sometimes be difficult to cut through the noise to *find* the information you need and *understand* it once you have found it.

Information architecture (IA) is a design discipline that is focused on making information findable and understandable. Because of this, it is uniquely well suited to address these challenges. IA allows us to think about problems through two important perspectives: that information products and services are perceived by people as *places made of information*, and that these information environments can be *organized for optimum findability and understandability*.

This first part of the book explains what IA is, what problems it solves, and how it can help you create more effective products and services. Part II and Part III will then show you how.

Let's get started!

CHAPTER 2

Defining Information Architecture

We say nothing essential about the cathedral when we speak of its stones.
—Antoine de Saint-Exupéry

In this chapter, we'll cover:

- A working definition (or four!) of information architecture
- Why it's so hard to point to something and say, "that's a great IA"!
- A model for effective IA design

If you're new to information architecture, at this point you may be wondering what this is all about. This chapter has answers for you! And if you have been working in one of the UX design disciplines for a while, you may be thinking, "But isn't information architecture about making sitemaps, wireframes, and website navigation menus?" Well, yes—these are important elements of information architecture design. But there is much more to this story! In this chapter, we'll give you a broader picture of what information architecture is—and isn't.

Definitions

Let's start by clarifying what we mean by information architecture:

1. The structural design of shared information environments
2. The synthesis of organization, labeling, search, and navigation systems within digital, physical, and cross-channel ecosystems
3. The art and science of shaping information products and experiences to support usability, findability, and understanding
4. An emerging discipline and community of practice focused on bringing principles of design and architecture to the digital landscape

Were you expecting a single definition? Something short and sweet? A few words that succinctly capture the essence and expanse of the field of information architecture? Keep dreaming!

The reason we can't serve up a single, all-powerful, all-purpose definition is a clue to understanding why it's so hard to design good digital products and services. We're talking about the challenges inherent in language and representation. No document fully and accurately represents the intended meaning of its author. No label or definition totally captures the meaning of a document. And no two readers experience or understand a particular document or definition or label in quite the same way. The relationship between words and meaning is tricky at best.[1] And here's the paradox of defining information architecture: by defining and clarifying semantic concepts, IA makes them more understandable and findable, but at a cost, because definitions are so imperfect and limiting at the same time. The definition of IA itself is a great illustration of this paradox.

We'll now descend from our philosophical soapbox and get down to basics. Let's expand on our definitions to explore some basic concepts of information architecture:

1 For a humorous perspective on the trickiness of the English language, see Bill Bryson's *The Mother Tongue: English and How It Got That Way* (New York: William Morrow, 1990).

Information

We use the term "information" to distinguish information architecture from data and knowledge management. Data is facts and figures. Relational databases are highly structured and produce specific answers to specific questions. Knowledge is the stuff in people's heads. Knowledge managers develop tools, processes, and incentives to encourage people to share that stuff. Information exists in the messy middle. With information systems, there's often no single "right" answer to a given question. We're concerned with information of all shapes and sizes: websites, documents, software applications, images, and more. We're also concerned with metadata: terms used to describe and represent content objects such as documents, people, processes, and organizations.

Structuring, organizing, and labeling

Structuring involves determining the appropriate levels of granularity[2] for the information "atoms" in your product or service, and deciding how to relate them to one another. Organizing involves grouping those components into meaningful and distinctive categories, creating the right contexts for users to understand the environment they are in and what they're looking at. Labeling means figuring out what to call those categories and the navigation structure elements that lead to them.

Finding and managing

Findability is a critical success factor for overall usability. If users can't find what they need through some combination of browsing, searching, and asking, then the system fails. But designing for the needs of users isn't enough. The organizations and people who manage information are important, too. An information architecture must balance the needs of users with the goals of the business. Efficient content management and clear policies and procedures are essential.

Art and science

Disciplines such as usability engineering and methodologies such as ethnography bring the rigor of the scientific method to the analysis of users' needs and information-seeking behaviors.

2 Granularity refers to the relative size or coarseness of information chunks. Varying levels of granularity might include journal issue, article, paragraph, and sentence.

> We're increasingly able to study patterns of usage and subsequently make improvements to our websites. But the practice of information architecture will never be reduced to numbers; there's too much ambiguity and complexity. Information architects must rely on experience, intuition, and creativity. We must be willing to take risks and trust our intuition. This is the "art" of information architecture.

Just Because You Can't See It, Doesn't Mean It Isn't There

One of the challenges people have with information architecture is that they can't easily point to it. How many times have you heard someone say, "Boy, that website's information architecture is really terrific!" or, "I can't find anything in this app! Its information architecture sucks!" Our bet is, not many. But the fact that you can't readily *see* the information architecture in things doesn't mean it's not there. As de Saint-Exupéry said, sometimes what is essential is invisible to the eye.

To illustrate, think of the game of chess. Perhaps the image that comes to your mind is of a chessboard like the one shown in Figure 2-1, with beautifully sculpted wooden pieces, and a goblet of brandy sitting near a flickering fireplace. That beautiful chessboard is a common instantiation of the game we call chess. However, chess is more than that. You could argue that what makes chess "chess" is a set of information structures that relate to one another according to predefined rules.

To begin with, chess has a taxonomy of pieces that represent army units: pawns, rooks, bishops, knights, kings, and queens. In play, there are two sets ("armies") of such pieces: "black" and "white." These armies face each other in a field that consists of an eight-by-eight grid of alternating light- and dark-colored squares. This field—the chessboard—creates a context (a "place") for the battle to take place.

Figure 2-1. A chess board with pieces in the opening position (image: http://bit.ly/opening_chess_position)

The different types of pieces can move and interact in different ways in this board; there are lots of rules that determine how the armies can interact. Differences in the pieces' range, scope, and numbers determine their relative worth to each army (Table 2-1).

Table 2-1. The different types of chess pieces, including their relative values and starting amounts

Name	Amount per army	Relative value
Pawn	8	1
Knight	2	3
Bishop	2	3
Rook	2	5
Queen	1	9
King	1	—

(The king is invaluable: its capture ends the game.)

So think back to the beautiful wooden chess set. If chess can be reduced to these basic information structures, perhaps you're suspecting that the wooden pieces and board are somewhat superfluous and that you should be able to play chess with many different types of sets. You'd be correct: in fact, chess can be played in many

different ways that do not involve carved wood—or any types of physical pieces—at all. For example, you may have heard of correspondence chess, which is played via postal mail using pen and paper (Figure 2-2).

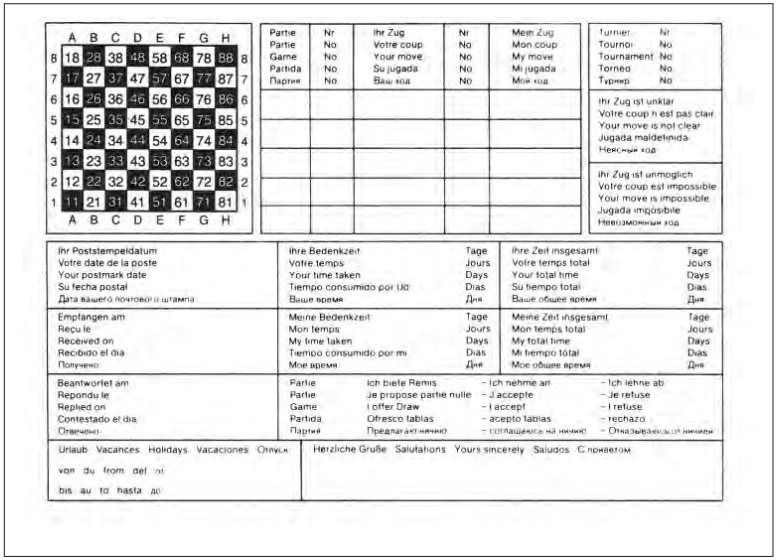


Figure 2-2. Correspondence chess postcard (image: Schach Niggemann GFDL, http://www.gnu.org/copyleft/fdl.html, or CC-BY-SA–3.0, http://creativecommons.org/licenses/by-sa/3.0/, via Wikimedia Commons)

Or perhaps you're more familiar with chess as a video game, an example of which is shown in Figure 2-3. This variant is played on a computing device with the board and pieces rendered as pixels on a screen, with the game mechanics adjusted to conform to the device's user interface particularities.

Figure 2-3. Deep Green chess, played on an iPhone with a touchscreen interface

Chess can also be played in a computer terminal console, with the most minimally symbolic user interface imaginable (Figure 2-4).

Figure 2-4. GNU Chess, played with a command-line interface

And of course, there are also countless variations of physical chess sets, ranging from our beautiful wooden set, to cheap "travel" sets with minimally rendered graphics on magnetic pieces (Figure 2-5), to the "Jewel Royale Chess Set" that is valued at almost $10 million.

These incarnations of chess are all physically very different from one another, yet they are all still chess. Why? Because they make possible and express the underlying information structures and rules of chess. Expressing and supporting these information structures is what *makes* all of these incarnations chess; their physical form and interaction mechanisms are merely matters of interaction or industrial design. In many ways, this abstract idea of chess is more "real"—but less tangible—than the physical (or virtual) chess sets that we interact with, because it is what makes chess different from other games.

Figure 2-5. Intense game of chess unfolding on a cheap magnetic travel set (image: http://bit.ly/magnetic_chess; cropped)

It's worth noting that nobody set out to explicitly create this "information architecture" of chess—the game, its piece types and rules, its lore, etc. have evolved over centuries. This is also true of the ways we've organized other information structures that afford understanding over time: it's only in retrospect that we can point to them and say, "that's a damned good information architecture!"

Toward a Damned Good Information Architecture

Users. Content. Context. You'll hear these three words again and again throughout this book. They form the basis of our model for practicing effective information architecture design. Underlying this model is a recognition that you can't design useful information architectures in a vacuum. An architect can't huddle in a dark room with a bunch of content, organize it, and emerge with a grand solution. It simply won't hold up against the light of day.

Websites, intranets, apps, and other information environments are not lifeless, static constructs. Rather, there is a dynamic, organic nature to both the information systems and the broader contexts in which they exist. This is not the old world of yellowing cards in a library card catalog. We're talking complex, adaptive systems with emergent qualities. We're talking rich streams of information flowing within and beyond the borders of departments, business units, institutions, and countries. We're talking messiness and mistakes, trial and error, survival of the fittest.

We use the concept of an "information ecology"[3] composed of users, content, and context to address the complex dependencies that exist in these information environments. And we draw upon our trusty Venn diagram (see Figure 2-6) to help people visualize and understand these relationships. The three circles illustrate the interdependent nature of users, content, and context within a complex, adaptive information ecology.

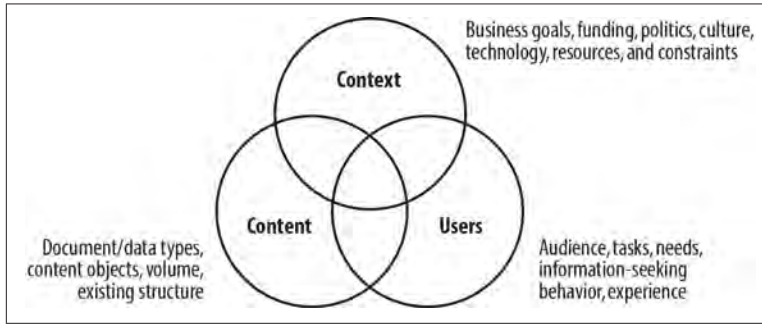


Figure 2-6. The infamous three circles of information architecture

In short, we need to understand the business goals behind the project and the resources available for design and implementation. We need to be aware of the nature and volume of content that exists today and how that might change a year from now, and we must learn about the needs and information-seeking behaviors of our major audiences.

3 For more about information ecologies, read *Information Ecology* by Thomas Davenport and Lawrence Prusak (Oxford: Oxford University Press, 1997) and *Information Ecologies* by Bonnie Nardi and Vicki O'Day (Cambridge, MA: MIT Press, 1999). Nardi and O'Day define an information ecology as "a system of people, practices, values, and technologies in a particular local environment."

Good information architecture design is informed by all three areas, and all three are moving targets. Users can vary in their attitude, demographics, psychographics, tasks and information needs, information-seeking behaviors, and more. Content can vary in quality, currency, authority, popularity, strategic value, cost, and more. And organizational context can vary based on mission, vision, goals, organizational politics, organizational culture, degree of centralization or autonomy, and more. The particular mix of variables differs from one information environment to another, and within the same environment it varies over time.

Even so, this is an oversimplified view of reality. Is it still useful? Absolutely. We've been using this model for 20 years. It's held up well in all sorts of environments, from global websites of Fortune 100 corporations to standalone intranet applications within small nonprofits. More importantly, we find these three circles incredibly helpful whenever we're confronted by a difficult question. After mouthing the trusty phrase "It depends"—as all smart practitioners of information architecture do—we develop our answer by deconstructing the question into three parts that coincide with our three circles. When asked what are the most important qualities that we should bring to the table, the answer becomes quite simple: some knowledge of users and their needs (which might come from exposure to human–computer interaction and a variety of other fields), content (think technical communication and journalism), and context (read a book on organizational psychology).

The three circles help with other tough questions, too, such as:

- What research and evaluation methods should we be familiar with?
- What kinds of people should be part of the team that designs the information architecture?
- What kinds of books and blogs should we read to keep up with the field and its practice?
- What should go into the IA strategy that we propose to a new prospect?

The answer to each starts with a balance among the three areas: users, content, and context.

Should technology have its own circle? Maybe. But we find that technology usually gets too much attention. Also, we increasingly find that much of what falls under the rubric of technology can be expressed within the "context" circle. After all, what technology brings to the table are new possibilities and constraints that give shape to the final product, and this is squarely within the realm of the context we're designing for.

Incidentally, we think it's important to have a good sense of humor about this stuff. Perhaps you've already figured this out. The work we do involves high levels of abstraction, ambiguity, and occasionally absurdity, and to some degree we're all still making it up as we go along.

If there's one thing that many years of information architecture consulting has taught us, it's that every situation is unique. We don't just mean that websites are different from intranets or that extranets should vary by industry. We mean that, like fingerprints and snowflakes, every information ecology is unique. The Toyota intranet is vastly different from that of Ford or GM. Fidelity, Vanguard, Schwab, and E*TRADE have each created unique online financial-service experiences. Despite all the copycatting, benchmarking, and definitions of industry best practices that have surged throughout the business world in recent years, each of these information systems has emerged as quite distinctive.

That's where our model comes in handy. It's an excellent tool for learning about the specific needs and opportunities presented by a particular project. Let's take a look at how each of our three circles contributes to the emergence of a totally unique information ecology.

Context

All digital design projects exist within a particular business or organizational context. Whether explicit or implicit, each organization has a mission, goals, strategy, staff, processes and procedures, physical and technology infrastructure, budget, and culture. This collective mix of capabilities, aspirations, and resources is unique to each organization.

Because of this, information architectures must be uniquely matched to their contexts. The vocabulary and structure of your websites and your apps is a major component of the evolving con-

versation between your business and your customers and employees. It influences how they think about your products and services. It tells them what to expect from you in the future. It invites or limits interaction between customers and employees. Your information architecture provides perhaps the most tangible snapshot of your organization's mission, vision, values, strategy, and culture. Do you really want that snapshot to look like that of your competitor?

The key to success is understanding and alignment. First, you need to understand the business context. What makes it unique? Where is the business today, and where does it want to be tomorrow? In many cases, you're dealing with tacit knowledge. It's not written down anywhere; it's in people's heads and has never been put into words. We'll discuss a variety of methods for extracting and organizing this understanding of context. Then, you need to find ways to align the information architecture with the goals, strategy, and culture of the business. We'll discuss the approaches and tools that enable this custom configuration.

As mentioned in Chapter 1, you also need to understand the contextual differences imposed by the channels that the user will be using to interact with your organization. Will they be experiencing your services primarily via apps on mobile phones, or via a website in a desktop-based browser? Both platforms have things they can do well, and things they can't. For example, smaller screens mean less space, which in turn implies shorter labels and navigation menus. Devices with small screens are also used at different times and places than those with larger screens. If your service will be used via more than one channel, you need to consider how these channels will overlap and interact with one another. All of these factors form part of the context that will shape your information architecture.

Content

We define "content" very broadly to include the documents, applications, services, schemas, and metadata that people need to use or find in your systems. To employ a technical term, it's the "stuff" that makes up your sites and apps. Many digital systems are heavily textual; among other things, the Web is a wonderful communication tool, and communication is built upon words and sentences trying to convey meaning. Of course, we also recognize it as a tool for tasks and transactions; a flexible technology platform that supports buying and selling, calculating and configuring, sorting and simulating.

But even the most task-oriented ecommerce website has "content" that customers must be able to find.

As you survey content across a variety of digital systems, the following facets will bubble to the surface as distinguishing factors of each information ecology:

Ownership
: Who creates and owns the content? Is ownership centralized within a content authoring group or distributed among functional departments? How much content is licensed from external information vendors? How much is produced by the users themselves? The answers to these questions play a huge role in influencing the level of control you have over all the other dimensions.

Format
: Websites and intranets have become the unifying means of access to all digital formats within many organizations. Databases, product catalogs, discussion archives, technical reports in MS Word, annual reports in PDF, office supply purchasing applications, and video clips of the CEO are just a few of the types of documents, databases, and applications you'll find on a given site.

Structure
: All documents are not created equal. An important memo may be fewer than 100 words. A technical manual may be more than 1,000 pages. Some information systems are built around the document paradigm, with the fully integrated document as the smallest discrete unit. Other systems take a content component or digital asset approach, leveraging some form of structural markup (e.g., XML or JSON) to allow management and access at a finer level of granularity.

Metadata
: To what extent has metadata that describes the content and objects within your system already been created? Have documents been tagged manually or automatically? What's the level of quality and consistency? Is there a controlled vocabulary in place, or have users been allowed to tag the content? These factors determine the extent to which you're starting from scratch

with respect to both information retrieval and content management.

Volume
: How much content are we talking about? A hundred applications? A thousand pages? A million documents? How big is the system?

Dynamism
: What is the rate of growth or turnover? How much new content will be added next year? And how quickly will it go stale?

All of these dimensions make for a unique mix of content and applications, which in turn suggests the need for a customized information architecture.

Users

The most important thing to know about users is that when we are talking about "users" we are talking about *people*. These are human beings with desires, needs, concerns, and foibles—just like you and us. We use the word "users" as shorthand to mean "the people who will use your information environment."

When we worked on the first corporate website for Borders Books & Music, back in the mid-1990s before Amazon became a household name, we learned a lot about how customer research and analysis was applied to the design and architecture of physical bookstores.

Borders had a clear understanding of how the demographics, aesthetic preferences, and purchasing behaviors of its customers differed from those of its main competitor, Barnes & Noble. It was no mistake that the physical layout and the selection of books differed significantly between these two stores, even within the same town. They were different by design. And that difference was built upon an understanding of their unique customer or market segments.

Differences in customer preferences and behaviors within the physical world translate into different information needs and information-seeking behaviors in the context of websites and apps. For example, senior executives may need to find a few good documents on a particular topic very quickly. Research analysts may need to find all the relevant documents and may be willing to spend several hours on the hunt. Managers may have a high level of industry knowledge but low navigation and searching proficiency. Teenagers

may be new to the subject area but skilled in handling a search engine.

Do you know who's using your system? Do you know how they're using it? And perhaps most importantly, do you know what information they want from your systems? These are not questions you can answer in brainstorming meetings or focus groups. As our friend and fellow information architect Chris Farnum likes to say, you need to get out there in the real world and study your "users in the mist."

Recap

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

- There's more than one way to define information architecture, and that's OK.
- Information architecture is not something you can easily point to; it is mostly abstract and exists below the surface, in the deep semantic structures of products and services. This is OK, too!
- Our model for practicing effective information architecture design considers three things: users, context, and content.
- The particular mix of variables changes not just from one information environment to another, but also for a single information environment over time.

As we mentioned in the introduction to Part I, IA is focused on making information environments *findable* and *understandable*. These are related, but different, objectives. In the next chapter, we'll look more closely at designing for findability. Onward!

CHAPTER 3
Design for Finding

I've had thank-you emails from people whose lives have been saved by information on a medical website or who have found the love of their life on a dating website.
—Tim Berners-Lee

In this chapter, we'll cover:

- Different models for how people look for information
- People's information-seeking behaviors
- How we learn about these behaviors

Information architecture is not restricted to taxonomies, search engines, and the other things that help users find stuff in an information environment. Information architecture starts with people and the reason they come to your site or use your app: they have an information need.

This is a truism, but there's more to it than meets the eye. Information needs can vary widely, and each type of information need causes people to exhibit specific information-seeking behaviors. It's important that you understand those needs and behaviors, and shape your designs to correspond accordingly. There is no goal more important to designing information architecture than to satisfy peoples' needs.

For example, if your information environment is a web-based staff directory, looking up a staff member's phone number is probably a very common information need among your users; in fact, this type

of need may describe most of your users' finding sessions. When confronted by such a need, people will likely perform a search, and you'd be wise to make sure your information architecture supports searching by name. On the other hand, if your product helps non-savvy investors learn about and select mutual funds for investment, your users may satisfy this need through some other means. They might benefit from a step-by-step tutorial, or they may wish to wander by browsing through categories.

Seeking something you know is there, like your colleague's phone number, is quite a different information need than learning about a topic like small-cap mutual funds, and your system's information architecture should be designed with those differences in mind. These kinds of needs lead to different information-seeking behaviors; not surprisingly, searching for something you know exists involves a very different behavior than browsing for the unknown. Distinguishing between these needs and behaviors and determining which are your users' highest priorities is an extremely valuable pursuit—it helps you determine where to invest your efforts and resources as you design your architecture.

The "Too-Simple" Information Model

There are different models of what happens when people look for information. Modeling needs and behaviors forces us to ask useful questions about what kind of information users want, how much information is enough, and how they actually interact with the architecture.

Unfortunately, "too simple" is the most common information model, and it's also the most problematic. It looks something like Figure 3-1.

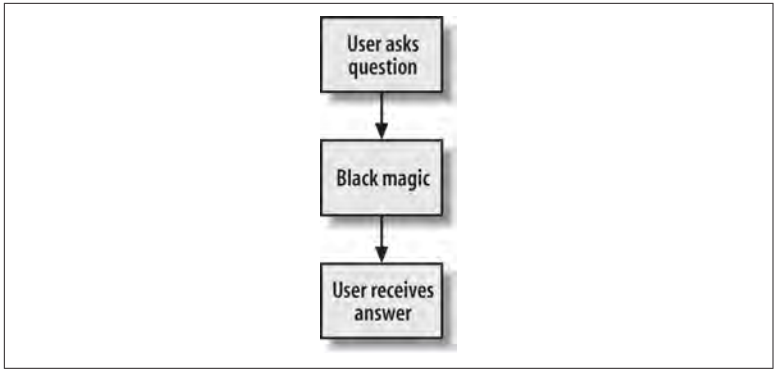


Figure 3-1. The "too-simple" model of information needs

Or, expressed as a simple algorithm:

1. User asks a question.
2. Something happens (i.e., searching or browsing).
3. User receives the answer.
4. Fin.

Input, output, end of story. This is a very mechanistic and ultimately dehumanizing model for how people find and use information. In fact, in this model, the user, like the site or app itself, is just another system—predictable in behavior, rational in motivation.

Why do we have a problem with this "too-simple" model? Because it rarely happens this way. There are exceptions—for example, when people know what they're looking for, as in the staff directory scenario. Here, users have a question for which there is a right answer, they know where to find the answer, they know how to state the question, and they know how to use the system to do so.

But people don't always know exactly what they want. Have you ever visited a website just to poke around? By exploring the site, you're trying to find information of a sort; you just don't know exactly what you're looking for. Even when you do, you may not have the language to express it: is it "skin cancer," or "melanoma"?

People often complete their efforts at finding information in a state of partial satisfaction or outright frustration. Example: "I was able to find information on synchronizing my iPhone, but nothing specific on syncing to Lotus Notes." Or, during the process of finding, they

may learn new information that changes what they're looking for altogether. Example: "I realized that a Roth IRA is ideal for me, even though when I started I was trying to learn about retirement plans."

We also dislike the "too-simple" model because it narrowly focuses on what happens while the user is interacting with the information architecture. The information need's context—all the related stuff that happens before and after the user ever touches the keyboard—gets left out. It also assumes an ignorant user who brings little, if any, prior knowledge to the table. So, the model essentially ignores any context for this scenario.

Finally, by oversimplifying, this model cedes so many great opportunities to understand what goes on in users' heads and observe the richness of what happens during their interactions with an information architecture.

This model is dangerous because it's built upon a misconception: that finding information is a straightforward problem that can be addressed by a simple, algorithmic approach. After all, we've solved the challenge of retrieving data—which, of course, is facts and figures—with database technologies such as SQL. So, the thinking goes, let's treat the abstract ideas and concepts embedded in our semistructured textual documents the same way.

This attitude has led to the wasting of many millions of dollars on search engine software and other technological panaceas that would indeed work if this assumption were true. Many user-centered design techniques carry this misconception forward, assuming that the process of finding is simple enough to be easily measured in a quantifiable way. So we think we can measure the experience of finding by how long it takes, or how many mouse clicks it takes, or how many viewed pages it takes to find the "right" answer, when often there *is* no right answer.

OK, enough complaining about this model. Let's take a closer look at information needs and seeking behaviors so that we can build better models.

Information Needs

When someone visits a website to find something, what does she really want? In the "too-simple" model, she wants the "right answer" to her question. Indeed, right answers are found from searching

databases, which store facts and figures and answer questions that really do have right answers, such as "What is the population of San Marino?" To many of us, database searching is the most familiar model of searching.

But digital systems store much more than highly structured data. Not surprisingly, text is the most common type of data stored, and text itself is made up of ambiguous, messy ideas and concepts. When we go to a website for advice on retirement investing, to learn about restaurants in Mendocino County, or to find out what's happening with the Manchester United football team, we are essentially looking for ideas and concepts that inform us and help us make decisions. The answer, if there is one, is an ambiguous moving target.

So back to the question: what do people want? Let's use the metaphor of fishing to get at the answer:

The perfect catch
: Sometimes users really are looking for the right answer. Let's think of that as fishing with a pole, hoping to hook that ideal fish. What is the population of San Marino? You go to Wikipedia or some other useful site that's jam-packed with data, and you hook in that number (it's 32,576, by the way, according to the latest estimate). And you're done, just as the "too-simple" model would have it.

Lobster trapping
: What about the times you're looking for more than just a single answer? Let's say you're hoping to find out about good bed-and-breakfast inns in Stratford, Ontario. Or you want to learn something about Lewis and Clark's journey of exploration. Or you need to get a sense of what sort of financial plans can help you save for retirement. You don't really know much about what you're looking for, and aren't ready to commit to retrieving anything more than just a few useful items, or suggestions of where to learn more. You're not hoping to hook the perfect fish, because you wouldn't know it if you caught it. Instead, you're setting out the equivalent of a lobster trap—you hope that whatever ambles in will be useful, and if it is, that's good enough. Perhaps it's a few candidate restaurants that you'll investigate further by calling and checking their availability and features. Or maybe it's a motley assemblage of Lewis and Clark stuff,

ranging from book reviews to a digital version of Clark's diary to information about Lewis & Clark College in Oregon. You might be happy with a few of these items, and toss out the rest.

Indiscriminate driftnetting
: Then there are times when you want to leave no stone unturned in your search for information on a topic. You may be doing research for a doctoral thesis, or performing competitive intelligence analysis, or learning about the medical condition affecting a close friend, or, heck, ego surfing. In these cases, you want to catch every fish in the sea, so you cast your driftnets and drag up everything you can.

I've seen you before, Moby Dick...
: There's some information that you'd prefer to never lose track of, so you'll tag it so you can find it again. Thanks to social bookmarking and collection services like Pinterest, it's possible to toss a fish back in the sea with the expectation of finding it again.

This fishing metaphor is helpful because it illustrates four common information needs. When you're hoping to make the perfect catch, you usually know what you're looking for, what to call it, and where you'll find it—this is called *known-item* seeking. An example is when you search the staff directory to find a colleague's phone number.

When you're hoping to find a few useful items in your traps, you're doing something called *exploratory seeking*. In this case, you're not exactly sure what you're looking for. In fact, whether you realize it or not, you're looking to learn something from the process of searching and browsing. For example, a user may go to his employer's human resources site to learn something about retirement plans that the company offers. In the process, he may encounter some basic information on specific types of plans, and then change his search to learning more about such plans. As he learns more about these plans, he shifts his search again to learning whether a simple or more complex plan is best for him. Exploratory seeking is typically open ended; there is no clear expectation of a "right" answer, nor does the user necessarily know how to articulate what exactly he is looking for. He is happy to retrieve a few good results, and use them as a springboard for the next iteration of the search. It's not always possible to definitively determine when exploratory searching is finished.

When you want everything, you're performing *exhaustive research*. You're looking for everything available on a particular topic, hoping to leave no stone unturned. In this case, the user often has many ways to express what she's looking for, and may have the patience to construct her search using all those varied terms. For example, someone who is trying to learn more about a friend's medical condition might execute multiple searches for "AIDS," "HIV," "acquired immuno-deficiency syndrome," and so forth. Again, there isn't necessarily a "right" answer. And in this case, the user must be patient enough to wade through more results than are typical with other information needs.

Finally, our failing memories and busy schedules continually force us to engage in *refinding* pieces of useful information that we've happened upon before. For example, while you're at work, you might surf for a few minutes and stumble on a great but long explanation of Django Reinhardt's guitar technique. Naturally, you won't read it now and risk losing your job. You'll refind it later instead, or use a "read later" service such as Instapaper to return to it at a more convenient time.

Figure 3-2 illustrates these four different types of information needs. They are by no means the only ones, but many of your users' needs will fall into these categories.

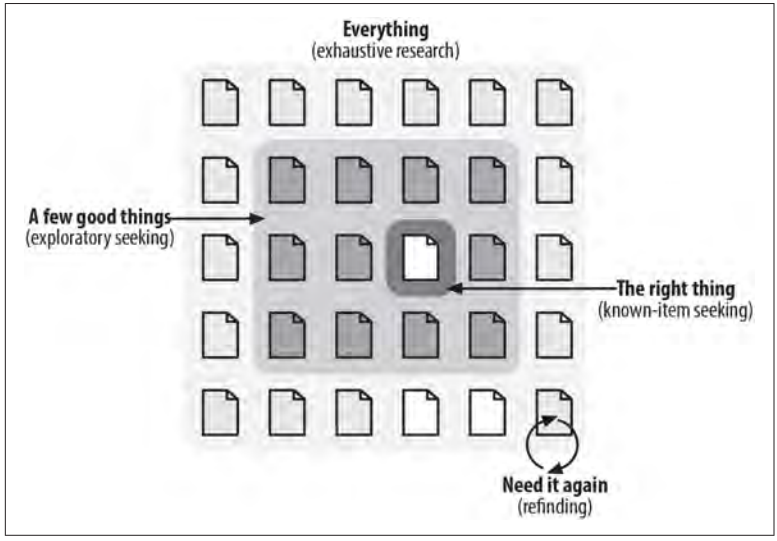


Figure 3-2. Four common information needs

Information-Seeking Behaviors

How do website users find information? They enter queries in search systems, browse from link to link, and ask humans for help (through email, chat interfaces, etc.). *Searching*, *browsing*, and *asking* are all methods for finding, and these are the basic building blocks of information-seeking behavior.

There are two other major aspects to seeking behaviors: *integration* and *iteration*. We often integrate searching, browsing, and asking in the same finding session. Figure 3-3 shows how you might search your corporate intranet for guidelines on traveling abroad. You might first browse your way through the intranet portal to the HR site, browse the policies area, and then search for the policy that includes the string "international travel." If you still didn't get your question answered, you might send an email to Biff, the person responsible for that policy, to ask exactly what your per diem will be while spending the week in Timbuktu. Let's hope your intranet's information architecture was designed to support such integration!

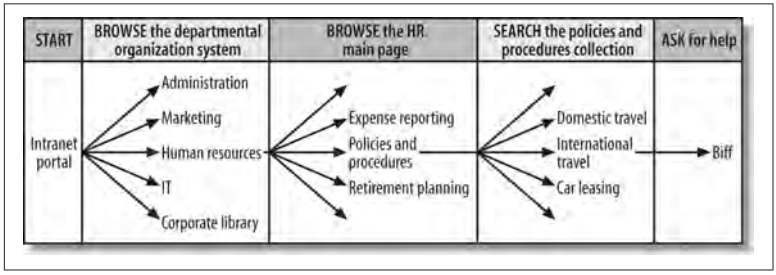


Figure 3-3. Integrated browsing, searching, and asking over many iterations

Figure 3-3 also illustrates the iteration you may go through during one finding session. After all, we don't always get things right the first time. And our information needs may change along the way, causing us to try new approaches with each new iteration. So, while you may have begun with a broad quest for "guidelines on traveling abroad," you might be satisfied to find something as specific as "recommended per diem in Timbuktu" by the time you're done. Each iteration of searching, browsing, asking, and interacting with content can greatly impact what it is we're seeking.

These different components of information-seeking behaviors come together in complex models, such as the "berry-picking" model

developed by Marcia Bates of the University of Southern California.[1] In this model (shown in Figure 3-4), users start with an information need, formulate an information request (a query), and then move iteratively through an information system along potentially complex paths, picking bits of information ("berries") along the way. In the process, they modify their information requests as they learn more about what they need and what information is available from the system.

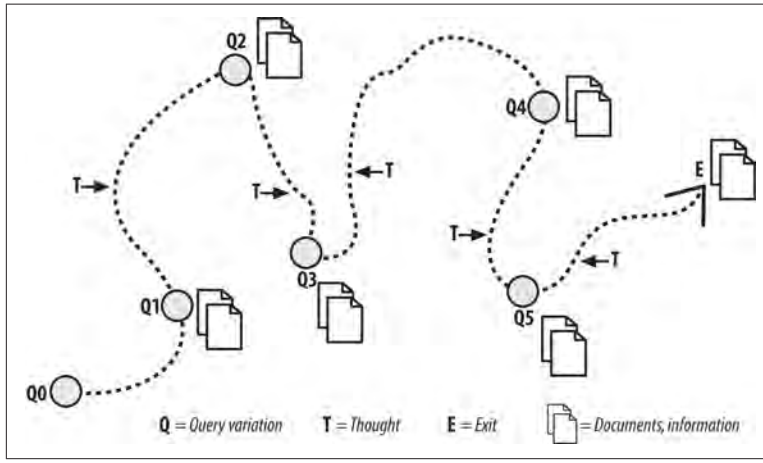


Figure 3-4. The "berry-picking" model of how users move through an information system

The berry-picking diagram looks messy—much more so than the "too-simple" model. It should; that's the way our minds often work. After all, we're not automatons.

If the berry-picking model is common to your users, you'll want to look for ways to support moving easily from search to browse and back again. Amazon.com provides one such integrated approach to consider: you can search within the categories you find through browsing, and you can browse through categories that you find by searching, as shown in Figure 3-5.

1 Bates's seminal paper, *"The Design of Browsing and Berrypicking Techniques for the Online Search Interface"* (*Online Review* 13:5, 1989, 407–425), is required reading for every information architect. She later expanded these ideas into a more comprehensive framework: see "Toward an Integrated Model of Information Seeking and Searching" (*New Review of Information Behaviour Research* 3, 2002, 1–15).

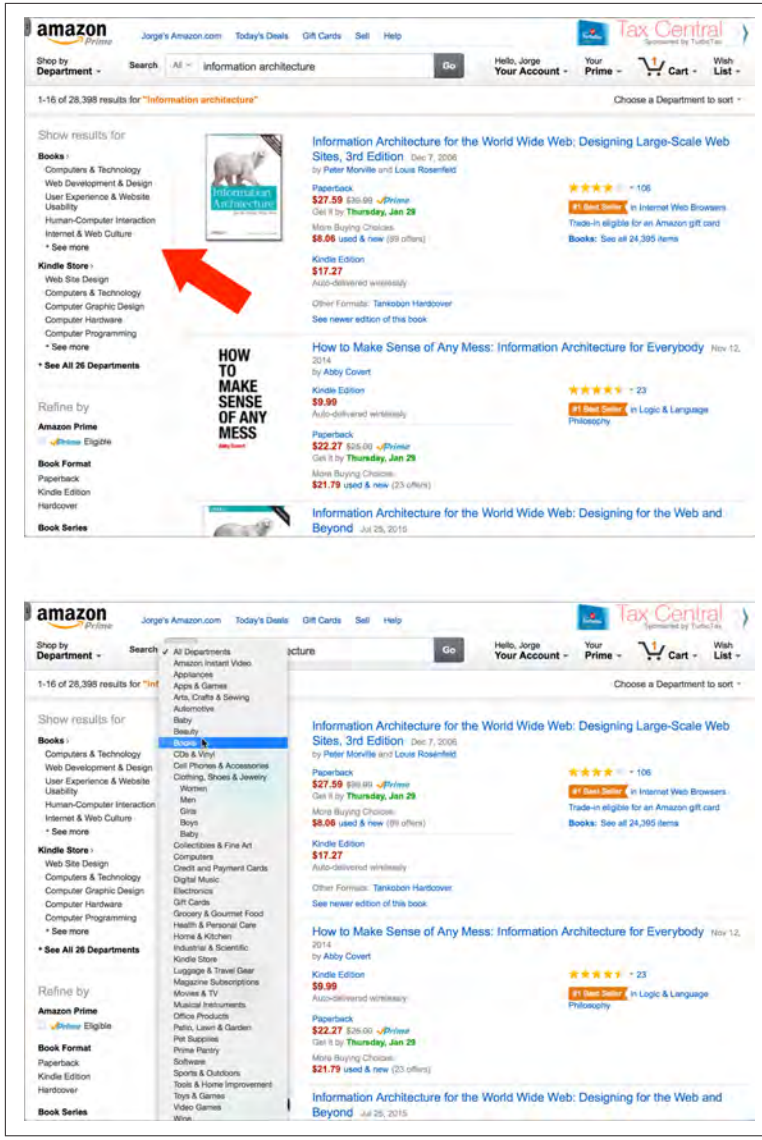


Figure 3-5. Browsing and searching are tightly integrated on Amazon.com

Another useful model is the "pearl-growing" approach. Users start with one or a few good documents that are exactly what they need. They want to get "more like this one." To meet this need, Google and many other search engines allow users to do just that: Google pro-

vides a command called "Similar pages" next to each search result. A similar approach is to allow users to link from a "good" document to documents indexed with the same keywords. In sites that contain scientific papers and other documents that are heavy with citations, you can find other papers that share many of the same citations as yours or that have been co-cited with the one you like. Delicious and Flickr are examples of sites that allow users to navigate to items that share something in common—in this case, the same user-supplied tags. All of these architectural approaches help us find "more like this one."

Corporate websites and intranets often utilize a "two-step" model. Confronted with a site consisting of links to perhaps hundreds of departmental subsites, users first need to know where to look for the information they need. They might search or browse through a directory until they find a good candidate or two, and then perform the second step: looking for information within those subsites. Their seeking behaviors may be radically different for each of these two steps; certainly, the information architectures typical of portals are usually nothing like those of departmental subsites.

Learning About Information Needs and Information-Seeking Behaviors

How can we learn about users' information needs and seeking behaviors? There are a variety of user research methods to consider—too many to cover in detail here—so we'll recommend a pair of our favorites: search analytics and contextual inquiry. Search analytics[2] involves reviewing the most common search queries on your site (usually stored in your search engine's logfiles) as a way to diagnose problems with search performance, metadata, navigation, and content. Search analytics provides a sense of what users commonly seek, and can help inform your understanding of their information needs and seeking behaviors (and it's handy in other ways, too, such as developing task-analysis exercises).

While search analytics is based on a high volume of real user data, it doesn't provide an opportunity to interact with users and learn more

2 For more on search analytics, read Lou's *Search Analytics for Your Site: Conversations with Your Customers* (Brooklyn, NY: Rosenfeld Media, 2011).

about their needs directly. Contextual inquiry,[3] a user research method with roots in ethnography, is a great complement to search analytics because it allows you to observe how users interact with information in their "natural" settings and, in that context, ask them why they're doing what they're doing.

Other user research methods you might look to are task analysis, surveys, and, with great care, focus groups. Ultimately, you should consider any method that might expose you to users' direct statements of their own needs, and when you can, use a combination of methods to cover as many bases as possible.

Finally, remember that your goal is to do your best to learn about your users' major information needs and likely information-seeking behaviors. A better understanding of what users actually want from your system will, naturally, help you determine and prioritize which architectural components to build, which makes your job much simpler (especially considering how many ways a particular information architecture could be designed). You'll also have great user data to help counterbalance the other drivers that too often influence design, such as budget, time, politics, entrenched technologies, and designers' personal preferences.

Recap

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

- IA starts with people and the reason they use your product or service: they have an information need.
- There are different models of what happens when people look for information.
- The most simple of these is problematic, because it doesn't accurately represent what actually happens when people have an information need.
- Information needs are like fishing: sometimes people know exactly what they're looking for, but often they're casting a wider net.

3 For more on contextual inquiry, read Hugh Beyer and Karen Holtzblatt's *Contextual Design: Defining Customer-Centered Systems* (Burlington, MA: Morgan Kaufmann, 1997).

- People act on these information needs through various information-seeking behaviors.
- There are various research methods that allow us to learn about these behaviors.

Now that we've learned about how people find information, let's move on to IA's second big goal: helping people *understand* information.