HCIN 620 Information and Interaction Design
The Last Word
Spring 2015

Let's give Bertin the last word on information architecture. First, though, there are a few unfinished things to say. We have been doing exercises and discussing various contributions to information architecture and need to enumerate some and demonstrate their connections.

Connecting Concepts

The following concepts: information scent, card sort, monitoring navigation, monitoring social networks, and flexibility of information representation, all come together to give us tools to build information containers. Let us briefly review them.

Information Scent. We discussed the propensity of content providers to make themselves easy quarry for information hunters. We reviewed an article that summarized practices for successful content as follows: visitors can get a quick snack, the menu changes frequently and predictably, and the dishes are easy to find, understand, and digest.

Understanding labels. We discussed card sorting as a means of understanding the labels people use to describe things of interest. We considered the issue of cognitive dissonance in the labeling of information containers and card sorting as a means to overcome it. We can connect the concepts of information hiding and labeling to see how labeling helps to limit information overload.

Learning navigational behavior. We have discussed the overwhelming influence of Google and its skill in rapidly changing information artifacts by learning the navigational behavior of its users. We have discussed the sharing of consumer information by commercial interests on the web through information brokers. We have discussed the local storage of tracking information on individual devices and some ways, such as LSOs (locally stored objects) used by commercial interests to thwart privacy preservation. We discussed the fact that many large websites, as well as many local network storage appliances, are better served by Google search than by any designed navigational aids. We considered the implications of that situation for expending effort to design navigational aids.

One thing I may have neglected to mention is A-B testing, a common practice where commercial websites compare the conversion rates of two features by randomly assigning visitors to one (A) or the other (B). This is a common way to choose design features. The conversion rate is the rate at which visits to a commercial site are converted to successful commercial transactions.

Social network influence on navigation. All commercial interests have recognized the significance of social networks and have devised ways to exploit social networks to influence navigation. Many navigational features in common use today are the result of specialists in a new field called network science drawing together research in many fields to understand human behavior and influence in networks. They use terms like betweenness centrality and network closeness. Major figures in the field include M.E.J. Newman, Stanley Wasserman, Albert Barabasi, Duncan Watts, and Lada Adamic.

Information structure. We have extensively discussed how and whether information is structured, using as a principle the degree to which human intervention is required to process information. We have discussed hierarchical and relational ways of organizing and storing information.

Flexibility of representation. We have touched on the notion that information structures are more or less amenable to change. Brittle structures may be symptomatic of technical shortcomings or may be symptoms of authoritarian governance. We discussed whether the speed with which we can modify an information artifact matters in a given context. We saw that we may put together an information artifact with little planning if we expect to take advantage of user behavior to improve it but that, if we can can not or will not change an information artifact after publishing it, we can not realize the value of understanding navigational behavior.

Design Patterns

We discussed several elements that information architecture authors have referred to as design patterns. To determine whether these elements deserve the label of *design pattern*, we must examine the coinage and past use of the term.

Design Pattern Definitions. The term *design pattern* is popularly used in many ways. Popular usage leads to an abbreviation of the original usage that may lose some of the original essence. Following are a few popular borderline uses of the term that I have seen that may just barely work but may erode understanding of the details.

- A general reusable solution to a commonly occurring problem within a given context
- A description (or template) for how to solve a problem, it can be used in many different situations
- An enumeration of the consequences of the use of the pattern in a given context
- Patterns are formalized best practices

Design patterns were first observed. Design patterns originated as a architectural concept by Christopher Alexander (1977). Alexander examined architecture from the standpoint of its value to a community of people in daily life. Alexander's ideas were largely ignored or rejected by architects but soon gained a cult following among computer scientists. Eventually his books became so popular outside architecture that they began to influence architecture.

A Pattern Language. Alexander describes it as a structured method of describing good design practices within a field of expertise. Interview The term was coined by Christopher Alexander and popularized by his book *A Pattern Language*. This book was followed by another book intended to explain the first book. Alexander has continued to try to explain the concept to this day.

Components of a Pattern Language. The Syntax describes where the solution fits into the larger design. The Grammar describes how the solution solves the problem. For example, "Balconies and porches which are less than 6 feet deep are hardly ever used."

An example of a pattern is a place to wait. The problem is that the process of waiting has inherent conflicts in it. The solution: In places where people end up waiting (for a bus, for an appointment, for a plane), create a situation which makes the waiting positive.

An example of a pattern is a useful cooking layout. The problem is that cooking is uncomfortable if the kitchen counter is too short and also if it is too long. Solution: To strike the balance between the kitchen which is too small, and the kitchen which is too spread out, place the stove, sink, and food storage and counter in such a way that:

1. No two of the four are more than 10 feet apart.
2. The total length of the counter—excluding sink, stove, and refrigerator—is at least 12 feet.
3. No one section of the counter is less than 4 feet long.

Computer scientists popularized design patterns. The Gang of Four (commonly abbreviated GoF) were among computer scientists seeking a basis to make code less arcane, more scientific and, above all, reusable.

One aspect of Alexander's description was so general that it seemed applicable to any field in which design plays a role. This key aspect was the notion of a *quality that could not be named* but that could be understood through experience—the quality shared by successful designs. Specific and non-obvious combinations of characteristics could support this quality.

Gang of Four book. *Design Patterns*, 1994, exploded on the software scene and propelled Alexander to greater fame at the same time as solidifying Object Orientation's place in mainstream software development.

The GoF argue that great writers use patterns, e.g., all of Shakespeare's plays were based on earlier, less successful plays or stories. The GoF refers to *tragically flawed hero* or *boy-meets-girl, boy-loses-girl* as patterns with infinite variety The GoF book serves two purposes, to tell what patterns are and to catalog 23 well-known patterns.

Gang of Four pattern definition. A design pattern is a description of communicating objects and classes customized to solve a general design problem in a particular context. (from the introduction to *Design Patterns*, 1994)

A pattern has four things.

1. Pattern name
2. Problem
3. Solution
4. Consequences

A pattern name is a tool. The pattern name must be good enough to become part of the design vocabulary. The pattern must be useful in conversation, documentation, and thinking. The GoF spent a lot of its time on the names of the 23 patterns in the catalog.

A problem may be of several kinds. The first kind includes basic design problems such as algorithm design. Another kind includes commonly occurring classes or object structures known to be problematic. A third kind includes lists of conditions that, if they occur together, create a generic problem.

A solution is a description of objects and classes. It is not a solution in a packaged sense. A solution is abstract, not implementation specific. A solution is a description of the elements of the solution (objects and classes). The description must identify

- Relationships between elements
- Responsibilities of elements

- Collaborations between elements

A consequence is a result or trade-off. The application of a pattern may resolve conflicts of various kinds, most often conflicts of space and time. To contemplate the use of a design pattern is to evaluate the design decision with awareness of the consequences. Consequences may have implementation issues, unlike the solution. If you feel tempted to talk about implementation, do so under the consequences banner instead of under the solution banner. Keep the solution a description, not an evaluation of itself.

Design patterns in information architecture. Information architects like Donna Spencer and the Information Architecture Institute use the term *design pattern* to refer to any observed pattern seen in numerous successful information containers, such as "tabbed menus", "hierarchies", and "hub-and-spoke". These are potentially useful structural elements. They simply don't rise to the level of design patterns in that they represent casual intuitive observation and nothing more. Alexander and the GoF spent years as master practitioners analyzing observed patterns and cataloging details, separating minutiae from valuable clues. I urge you to build on casual, intuitive observations and to do more. As you practice design, you may attain insights enabling you to develop genuine design patterns. But it will not be as easy as casual intuitive observation.

Magic fairy dust. A pioneer of HCI, Gary Olson, once told me that his main interdisciplinary frustration was that practitioners in other fields often wanted him to sprinkle magic fairy dust on their products. As HCI practitioners, you must avoid the complementary trap: don't mistake the work of Alexander and the GoF as magic fairy dust that can be sprinkled on your information architecture.

Bertin's Theory

It should not be surprising that the last word goes to a cartographer. This is one of two fields, the other being social network analysis, with the most influence on building information containers today.

Theories. The word theory is used to mean many things. To obtain a useful definition, it might be helpful to consider what Stephen Hawking says about theory in *A Brief History of Time* on page 7.

> A theory is a good theory if it satisfies two requirements. It must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements, and it must make definite predictions about the results of future observations. For example, Aristotle believed Empedocles's theory that everything was made out of four elements, earth, air, fire, and water. This was simple enough, but did not make any definite predictions. On the other hand, Newton's theory of gravity was based on an even simpler model, in which bodies attracted each other with a force that was proportional to a quantity called their mass and inversely proportional to the square of the distance between them. Yet it predicts the motions of the sun, the moon, and the planets to a high degree of accuracy.

Note that Hawking is giving what we might call a normative definition, telling us what a theory *should* be. There might be plenty of things parading around under the theory banner that could be classed as bad theory. Note also that Hawking requires that a theory both describe and predict. Bertin does both of these things, with a detailed description of graphical information and using a concept called efficiency to predict outcomes.

Efficiency. Bertin defines efficiency as a measure of how quickly the process of reading a graphic can be completed and defines the process in some detail under the influence of (then) contemporary philosophy. Let's read some information graphics now to get an idea of what this might entail.

It is important to understand that the above definition and review of information graphics is meant as scaffolding on the way to understanding efficiency. The precise definition of efficiency, quoted from Bertin, is *If, in order to obtain a correct and complete answer to a given question, all other things being equal, one construction requires a shorter observation time than another construction, we can say that it is more efficient for this question.* This is a measurable, precise definition. Bertin characterizes it as implementing Zipf's notion of mental cost.

Efficiency alone could be abused, so Bertin augments with some additional concepts. How do you think efficiency could be abused? What processes could be described to augment efficiency? (See Semiology of Graphics, p. 166 for an example.)

Categories of information graphics. Bertin describes four categories of graphics: diagrams, networks, maps, and symbols. It is the use of the last three categories, coupled with the detail in Bertin's theory, that persuade me that here is a good foundation for describing information architecture.

Symbols. Symbols establish a correspondence between a single element on the graphic plane and something external to the graphic, through immediate recognition. Symbols are used extensively in information architecture and Unicode offers a flexible way to represent 113,000 currently defined symbols in type, along with a plan for annual additions with a limit of about 1.1 million symbols using the current international agreements.

Here I am using the word flexible to mean that typographic glyphs are represented by curves that can be completely described by a short series of numbers in a specified order. If you have used Bezier curves in Photoshop, you know how to determine very complex shapes by specifying the location of a few *control points*. This same principle allows for device independence, arbitrary scaling of size, advanced algorithms for micro-kerning and other processes that depend on the relative locations of glyphs. Thus, typographic symbols are ideal vehicles for multimodal information representation in handheld, wrist worn, or desktop display.

Diagrams. Briefly, Bertin calls diagrams the constructions on a plan where correspondences occur between one component and another. I don't want to say more here because diagrams are more the domain of information content than navigation.

Networks. Bertin describes networks as the correspondence on the graphic plane between the divisions of one component. The underlying component is usually people in a social network but is usually represented as emails, texts, phone calls, links, or anything else that points from one object of a certain kind to another object of that kind.

Bertin's advice about the representation of networks dates from the 1960s, before the dawn of social network analysis. Nevertheless, he specifies the process of portraying a network that highlights the problem that still vexes practitioners today. First, establish the correspondences and second, identify the representation of the correspondences with the fewest intersections. In Bertin's book, he displays networks only a dozen or a few dozen nodes. Most of the network representations of today suffer from the production, in the first step, of correspondences numbering in the billions between divisions (people or websites) numbering in the hundreds of millions.

Plenty of algorithms have been developed to accomplish step two of the above process. Many are inventoried at graph drawing.

Bertin claims that any network may be converted to a diagram by the addition of one or more components. He gives the example of speakers and listeners as two components and demonstrates their portrayal with a bipartite graph and an incidence matrix.

Maps. Maps are exactly like networks except that a geographic order is imposed. The process for creating a map requires the reproduction of the geographic order first, and recording the correspondences second.

Although the network may be easily converted to a diagram, the map introduces a geographic ordering to the graphic plane that is disturbed by the addition of another component. Thus, I suspect that Bertin would call the graphical constructions based on the famous NYC subway maps diagrams rather than maps. In most cases, these constructions only superficially resemble a map in the sense that no geographic ordering occurs. Nevertheless, the notion of a map and the rules of cartography are deeply appealing to people, perhaps because of our hunter-gatherer past.

The lack of flexibility in maps is a key feature. The fact that additional components disturb the geographic ordering renders maps useful for navigation but diminishes their navigational usefulness if they are transformed into diagrams. This characteristic explains both the advantage and limitation of map-like navigational aids in information artifacts. If you map your information architecture as a flow chart, for instance, do you impose a geographic ordering on the 2D plane? If you do, you drastically limit the flexibility of the flow chart to represent abstract concepts but may enhance its use in describing flow.

Do you regard bread-crumb trails as maps or networks? Every single bread-crumb trail I have ever seen on a website follows a horizontal left older right newer path. Every single one I have ever seen in a menu follows a vertical down older up newer path. Each crumb offers only a click to return to it or perhaps a tooltip to provide a more complete name. The bread crumb trail has defied HCI researchers trying to evaluate its effects. Might Bertin's theory help us to understand what user studies have not?

Simplification. Recall my earlier question about adjuncts to efficiency. The main adjunct in my opinion is an exhaustive depiction of what it means to simplify information using retinal variables at the most detailed level possible. By far, the greatest value in *Semiology of Graphics* is, in my view, its scope. Much of the book contains the tools to describe the mechanisms of simplifying information. These are the retinal variables and their operation in a 2D plane. It is only by working extensively with these variables that you can build information artifacts that are more than randomly navigable.