Designing with the User in Mind a Cognitive Category Based Design Methodology

Joseph Kramer[1] and Sunil Noronha[2]

[1] IBM Research, Hawthorne, NY, USA
kramerjo@us.ibm.com
[2] Yahoo Inc., Sunnyvale, CA, USA
noronha@yahoo-inc.com

Abstract. To design products and experiences that are highly intuitive and resonate with their target users the designer must have an accurate understanding of those users 'mental models'. New research in cognitive science, in particular in the area of cognitive category theory, provides clues how to better elicit and apply mental models in design. The resultant outcome is guaranteed to be more natural and understandable to its users. In this paper we will briefly review the cognitive science research and describe our resultant empirically grounded concept and definition of a `mental model'. We then explain how we use the mental model and related design principles to build intuitive designs.

Keywords: Mental model, psychology, cognitive category, design method.

1 Introduction

The design of an intuitive and familiar experience requires the mental model of the target audience to be embodied in the experience design. This is the reason why, for example, the folder icon is universally used as a metaphor in software interfaces as the container for electronic files. The folder mimics the user's real world mental model where paper documents are stored in files and folders. A user is then able to transfer their real world knowledge and experience to inform their expectations of the affordances provided in the virtual world.

Unfortunately we frequently encounter design errors that violate users' mental models. Figure 1. is an actual screenshot from a search engine serving millions of users, for the query "lady gaga". The module leads off with a big news article about the musician. We observed users skip past that article and then click on the second title under Related Stories: "Lady Gaga takes on John McCain…". And then the users were stumped─ nothing seemed to happen when they clicked on the link!

Actually something did happen─the big title, "Lady Gaga releases video message" got replaced by "Lady Gaga takes on John McCain". I.e., the module was designed to replace the main article at the top by whichever Related Story was clicked by the user. However, the user didn't notice that. Their eyes had moved down, past the

A. Marcus (Ed.): DUXU 2014, Part IV, LNCS 8520, pp. 152–163, 2014.

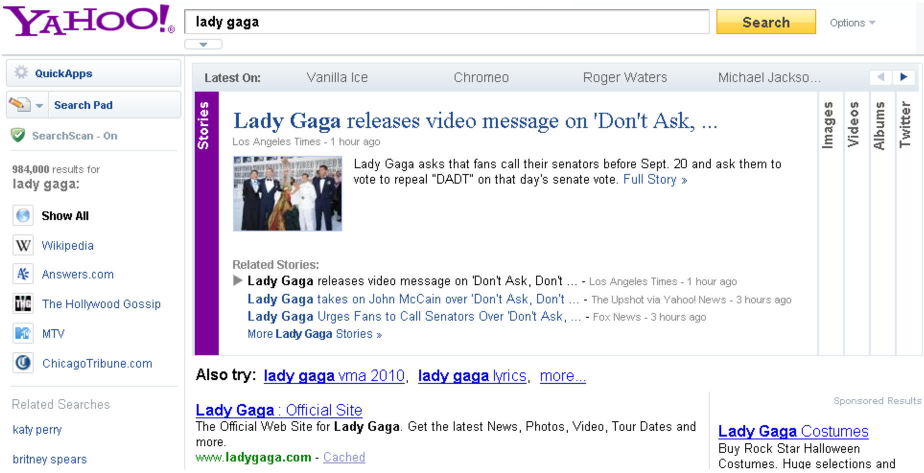

Fig. 1. A design error

main title, which they weren't expecting would change. And they were expecting that a click on a blue-link article would open the article.

Unfortunately the designer had a different mental model. They viewed the module as highlighting a news story. When the user clicked on another title, the designer wanted to highlight that story, and therefore switched the two stories.

This module was an expensive failure—it was actually built and deployed into production—and was taken down a few months later. However the amazing part is that a very simple technique could have prevented the error; this module would not even have passed beyond a whiteboard sketch if the designer had employed the techniques we present in this paper. (See the View Design section.).

Capturing or eliciting an audience's mental model is thus instrumental in most of the design tasks performed by user experience professionals. Given that issues of item naming, classification, grouping, hierarchical structuring, navigation, understandability, `findability', brand promotion, etc., are all frequent concerns, in this paper we will focus on the elicitation, capture and application of mental models to support design. We will briefly review the latest research in cognitive science, in particular in the area of cognitive category theory and describe the empirically grounded concept of a mental model as developed in that field. We then explain how we use them and related design principles in all phases of design.

2 Background and Literature Review

2.1 Mental models in HCI and Traditional Psychology

In HCI the term 'mental model' is used rather loosely. For example mental models have been defined as a set of beliefs about how a system works and humans interact with systems based on these beliefs [1]; any type of mental representation that develops during the interaction with the system and enables and facilitates interaction with the system [2]; what the user believes about the system at hand [3]; an affinity diagram of user behaviors [4].

In psychology the term "mental model" was first mentioned by Craik as being "small-scale models of reality" that people carry in their minds to anticipate events, to reason, and to underlie explanation [5]. Subsequently Johnson-Laird [6] used the term to describe the process which humans go through to solve deductive reasoning problems and Gentner & Stevens [7] proposed that mental models provide humans with information on how physical systems work.

All those definitions of `mental model' are perfectly fine for their original purposes. However our objective is the following. We want to create designs that are extremely intuitive because the design `reflects the way the user thinks'. In other words, the design `speaks the user's language' or `mirrors what is in the user's mind'. This is the crucial design principle of mimicry: if the design only communicates in terms of "whatever is in the user's mind", the user will completely understand every part of the experience very easily.

This requires us to know "what's in the user's mind", whatever that means in a formal (psychological) and literal sense. We therefore define a "mental model" as simply a formal representation or documentation of the content in a user's head.

We are thus referring to a *literal dump* of everyday-knowledge that is inside the user's head, e.g., "my home is 5 miles from my office". We are *not* referring to the user's hypotheses about how some system work works, e.g., "what's your mental model of how a search engine works?" Nor are we referring to affinity based groupings of user activities.

In other words, for us the notion of a `mental model' is an empirical one: it's whatever biology tells us about the actual contents of the human brain. This introduces a rigor into the design process that is not present with the other definitions of `mental model'. Now the design task is to look inside the user's head, in a literal sense, and mimic the contents.

So what does empirical research tell us about the contents of human minds? What are the "units of content", and what is the "structure" that relates those units?

This has been the subject of much research in psychology, and the first generation of this work was based on weak experimental methods, yielding concepts such as "attitudes", "values", etc., which is of largely theoretical origin and has low validity.

However research in "second generation cognitive science" is grounded in biological experiments in neuroscience, and has produced major breakthroughs in our understanding of the "mind" and its contents, well summarized by Lakoff and Johnson [8].

2.2 New Research into Mental Models

This research recognizes the fundamental impact of the `mind' being physically situated in the body, i.e., recognizing that the "mind" is essentially a piece of fiction, just a way of describing the functioning of the brain, a physical entity that can be subjected to laboratory experiments in neuroscience. Those experiments yield insight into the fundamental `information' structures being processed by the brain.

The primary mental structures so identified include cognitive categories, emotions, images, scripts, and so on. This area of study is very recent, and complete models of mental mechanisms are not yet available; however, the available components are already very useful for design.

In this paper we focus only on the mental structures known as cognitive categories. Cognitive categories roughly correspond to our everyday notion of "objects", e.g., "cars" and "cats" are cognitive categories, but "beautiful" is not. Thus cognitive categories provide the basic foundation for how the brain 'sees' the world.

There are several types of cognitive categories . Most people have no problem distinguishing cats from elephants, but have difficulty distinguishing one species of elephant from another. In other words, some types of categories, called "basic-level" categories are fundamental to human perception and most human knowledge is organized in terms of these categories. They are the most important ones for design.

The importance of the "basic-level" stems from the following properties [9];

1. It is the level at which most of our knowledge is organized
2. It is the level first named and understood by children
3. It is the level with the most commonly used labels for category members
4. It is the level at which category members have similarly perceived overall shapes
5. It is the level at which subjects are fastest at identifying category members
6. It is the highest level at which a single mental image can reflect the entire category. (We call this the "image marker rule")
7. It is the highest level at which a person uses similar motor programs for interacting with category members. (We call this the "action marker rule").

The first two points have a huge implication for design: they imply that the primary units of design for the general population should be basic-level cognitive categories. I.e., all objects that are surfaced in the design should correspond to this level if we wish to maximize intuitiveness and ease of use.

This point is reinforced by the other properties, each of which further increases the value of basic-level cognitive categories as the fundamental units of design. E.g., point 4 implies that all objects that correspond to a given basic-level category can be illustrated by a single picture; this has huge benefits when illustrating a product catalog for example. Point 5 implies that user effort is minimized at this level. Points 6 and 7 give us a pair of powerful design tools that we discuss later in this paper.

Another valuable aspect of cognitive categories pertains to the relationship-structure between categories. For reasons of space, we refer the reader to [8] for the theory and touch upon the design implications briefly in the section on View Design

Returning to our goal in this section of formalizing the notion of a "mental model", we now define it as follows:

> *A 'mental model' is a representation of the cognitive categories in a user's head, along with relationships between those cognitive categories. One way to represent a mental model on paper is to draw a graph comprising a node for each cognitive category, and an edge for each relationship linking cognitive categories.*

Anyone familiar with software engineering will immediately think of the UML Class Diagram, which is indeed a perfectly suitable tool for capturing a mental model─as long as other baggage from object-oriented programming is left out. It is important to recognize that this is meant to be a psychological diagram: it pictures the literal contents of the user's head, not a software system. Therefore the only objects permissible in a mental model diagram are those elicited from the user.

3 A Cognitive Category Based Design Methodology

Over many projects we have honed a systematic approach to product design that not only reduces a complex challenge to relatively easier steps, but almost guarantees that the resulting product will be useful, highly intuitive, and easy to use. The methodology makes heavy use of scientific principles, especially cognitive category theory. Parts of this methodology have been documented elsewhere but we briefly recap it here, because a firm grasp of the big picture makes it easier to dive into the technical details of each step and relate the steps to each other.

From afar these steps may sound familiar ─ use cases, mental models, etc.─ however there is an unusual engineering-like rigor and connectivity between the steps, arising out of the grounding in cognitive category theory, which seems to be new to most designers. We have rarely encountered anyone actually practicing the specifics described here unless they have been trained in this methodology.

Therein also lies the value of this methodology. Over numerous projects at Yahoo, IBM and elsewhere, we've observed that the majority of design errors that we've encountered were due to the designer or product owner omitting one of these steps, or doing it partially. When a designer has not trained themselves to instantly recognize the nature of an object, its relationships to other objects, and the corresponding design implications, they often miss errors that are instantly obvious to someone trained in these methods. In fact when we explain how to observe this structure and to exploit it, the techniques seem so simple and the errors so obvious that it sometimes triggers a sheepish reaction. We've observed that the learning curve for these techniques is very short, and most designers and product managers `get it' very quickly. This is especially true of people who have had some engineering training, and they usually express a lot of delight in the power of the techniques.

The following section will provide an overview of the four major design phases, and the subsequent sections will do a deep dive into each phase. Although this paper is meant to be primarily an aid to designers wanting to apply such techniques, we will also dive into the psychology underpinning these techniques for the benefit of researchers who want to understand why these techniques work as well as they do.

3.1 The Four Primary Phases of Design

Figure 2 is a birds-eye view of the four primary design steps. The primary function of these steps is to guarantee that a product will be `successful' in that sense that people will want to use it, it genuinely meets a need, and is highly intuitive and easy to use.

Those four steps are of course an over-simplification; there is a lot more to great product design than making a product useful, usable and intuitive. For example: how to build the product in a way that that people actually enjoy it and explicitly respond to it emotionally, and the product is perceived to be something truly different from its competitors. However the four primary steps mentioned above here are the foundation of any design project and in our experience we've seen designers get a tremendous amount of value from focusing on these steps alone.

The first step is about unearthing user needs and documenting them. It sounds straightforward, yet it's surprising how often product owners fail right at this step. When we have examined a product that failed and was shut down a few months after it launched, we have often found that the product had never documented a credible use case to begin with.

By "credible", we mean that the user needs must be (a) elicited from a typical user describing an actual situation or problem in their life and (b) documented entirely in the user's language. We have often encountered product requirements documents violating those two rules, typically written by a product owner who felt that the `use cases' they wrote were perfectly plausible to themselves. However we've learnt by watching failed and successful projects that the above criteria are hard rules. When they are violated there is a high risk of the product being designed for a `problem' that doesn't exist, leading to subsequent failure in the marketplace.

In theory one can argue that user language is not necessary for describing scenarios, since users may not be aware of their own need, or because a truly new product may create a need that previously didn't seem to exist. However in practice we have observed that the absence of user language in use cases is a marker that no one has actually verified that the product addresses a real need, and predicts product failure. Therefore we have learnt to treat it as a red flag indicating high project risk.

In addition to ensuring that we are addressing genuine user needs, the principle of documenting literal user language has another benefit: it provides the primary data input for our second step, "mental models".

The second step is to document which objects `manifest themselves' in the user experience, and how these objects relate to each other. Again the crucial rule is that these objects must exist in the target users' heads—i.e., we cannot concoct new objects as an object-oriented computer programmer might do. For example, a "cluster of news articles" is not a valid object, even though they might be the typical output of clustering algorithms. The typical reader of news articles doesn't think in terms of "clusters", but thinks in terms of "topics". The latter might seem superficially like a `cluster', but is actually far more restricted—a user can readily select from list of common `topics' which ones interest them, but would struggle to deal with `clusters'

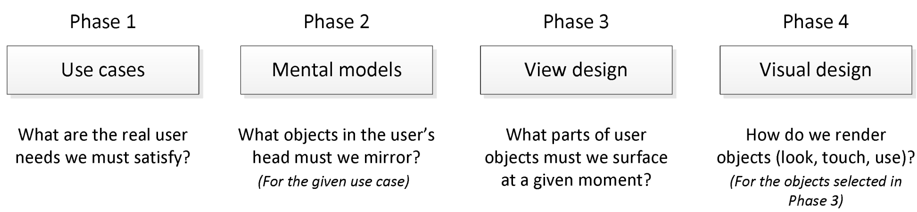

Fig. 2. The four phases of design

In order to determine which objects to surface within the user experience, we systematically walk through all the use cases from Step 1. All the important nouns in the user-stories are extracted as `user objects'. No other objects are introduced. That is why it was so important to elicit real use cases and document them in literal user language. The literal-language rule protects us from injecting unintuitive content and structure into the design.

The third step, view design, corresponds to what is frequently called `wireframing' but again with a discipline that does not permit arbitrary wireframes. In this step, each object from the user's head, which was elicited in Step 2, is now represented by a sketchy outline that specifies what attributes of the object are visible at a given point in the user experience. This step is not about how the object looks; colors, sizes and other visual elements are not specified. Only wireframe structure and visibility is decided. The wireframe structure is set up to mimic that of the user object model. The choice of which attributes to show is determined by walking through the use cases to find out which details of each object must be present in order to successfully complete the use case. This is done for all the use cases, and the attributes that are most frequently needed are selected for preferential treatment.

Clearly, this is in contrast to how typical wireframing is done. Most designers jump straight from user stories to sketching out wireframes; this is often viewed as a free-form construction or *creative* process where anything be sketched out. Instead in what we have described above is the wireframes are *derived* from the user-object model in a systematic way. The user's mental model thus strongly constrains exploration of the design space to the `set of good designs', tacitly eliminating designs that would be unintuitive because they would not match the user's mental model.

The final step, visual design, determines exactly how each object looks, down to a pixel. The primary function supported in this step is recognizability: with a quick glance a user must be able to recognize which object they're dealing with. In other words, visuals that are more `typical' of the object are better than atypical renderings. This step also achieves several other secondary objectives, such as aesthetics, conformance to standards, brand recognition, etc.

Considering all four steps together, it should be obvious that there are sequential dependencies between the steps. We cannot determine the right objects to surface unless we have first documented the use cases correctly. We cannot determine the correct structure of the wireframes unless we have elicited the intrinsic properties of the objects they represent. We cannot determine what kind of visuals to present unless we know what objects are represented by the visuals, and how much information we are required to present within a given space. Therefore, although it is possible to jump back and forth between steps to some degree, we really do need to perform them in the above sequence in order to determine the best possible design.

This point about sequential dependencies is important because it helps us recognize that the above methodology is not a "Process" that we're recommending for its own sake. In fact there is a lot of resistance to waterfall-like processes in `agile' product development paradigms, because unnecessary sequentiality can lead to slow execution; there is often a preference for rapid iteration as the primary means to figuring out the best design. Rather, what we're pointing out is that there are intrinsic technical dependencies between the steps, much as one cannot install the roof before building the rest of the house. So the 4-step design methodology is not about executing a `process'; rather it is about respecting the intrinsic `logic of design'. The sooner we figure out that logic, the more time we spend exploring the best designs.

3.2 The Role of Cognitive Categories

The above outline briefly touched upon cognitive categories aka "user-objects" in each of the four steps. We now take a closer look at how this grounding in psychology increases the quality and robustness of designs.

Use Cases. Revisiting Step 1, eliciting user needs, it follows that use cases must be captured in their everyday language, else we risk departing from the basic-level to more abstract, unusable cognitive categories. Sometimes however well intentioned designers introduce their own idealized or proposed means by which users could complete their tasks. The language introduced rapidly becomes part of the design team vernacular and ends up in the finished product, all whilst representing alien concepts that do not exist in the user's mental model.

User-objects, aka `Mental Models'. Step 2, documenting the user's mental model involves compiling a list of basic-level cognitive categories, when designing for the general population. Other cognitive categories may be used when the product will only be used by a niche customer segment. E.g., collectors of antique furniture may have a knowledge of many different types of chairs at an almost `basic level', with a distinctive image springing to their mind when you say "Queen Anne chair".

In this step, the most common error we have seen is introducing objects into the design that do not correspond to a user's basic level. The usual cause is that the product manager, marketer and designer understand the product domain differently from the user, because the former are intimately familiar with the `sausage-making' involved in creating the product and have developed their own technical vocabulary.

When a designer then jumps straight from use cases to wireframing and skip formal documentation of the assumed mental model, they inject non-user objects that subsequently weaken the design.

In addition to documenting a list of the cognitive categories that manifest themselves in naturalistic conversations with target users, Step 2 is about documenting the attributes of these cognitive categories and the actions that users do with them. It is common for an object to have numerous attributes, some concrete and some abstract, and not all attributes are equal from a user's perspective. For example, a TV has a size and a black-screen level, and the size is critical since it affects not only how great the image feels but also whether the TV fits with rest of the user's furniture.

The most important attributes of a cognitive category are the ones that define it, because they give an object its identity and help users instantly recognize it when they encounter it. Central to this are the image and action marker rules.

The Image Marker rule asks the question: When I say the word <chair>, does an image instantly spring to mind? If the answer is yes, the object is probably a basic-level category, and the image is a definitional image which we should strive to use to represent the object.

The Action Marker rule asks the question: When I say the word <chair>, does an action instantly spring to mind? Do you instantly know what to do with the object? If so, the object is probably a basic-level category, and the actions define the object, helping give it its identity.

When both these rules produce a "Yes" answer, we know we have in our hands the right unit of design. We also have the right image to use in the design, and the minimal set of key actions that we need to support everywhere on the given object.

Sometimes only one of these rules turns out to have a "Yes" answer, in which case we are not quite at the basic-level, but might still have in our hands a usable object.

Because these two rules are so trivially easy to apply, they are among the most powerful everyday design tools we have encountered. More often than not, when designing for a general population we can apply these rules from our armchairs, and therefore within minutes cut through the clutter of confusing hypothetical objects and abstract concepts. The subsequent clarity of design is striking.

View Design. Since this step essentially derives wireframes from the previous step's documentation of the user's mental model, cognitive categories end up playing a key role in determining what content gets displayed during the user experience. No object, attribute, or action appears that has not been present previously in the mental model.

Furthermore, this step adds a focus on making sure that the right content is available at the right time during each moment of the user experience. Most cognitive categories have numerous attributes, and showing too many of them can overwhelm the user and needlessly clutter the experience. Too few and the user cannot actually complete the task they set out to do. The attributes and actions are prioritized based on their `definitional' nature (i.e., they are core to identifying the object) and their utility for the given use cases. The image and action marker rules mentioned above are useful guides. Frequency of use—what is the estimated probability that users will need the attribute?—is the other primary guide to optimizing this selection process.

Often different contexts (points in the user experience) will require the same object but with different attributes. E.g., when presenting the initial page of a product catalog, it may suffice to just mention the category name "TVs", i.e., a 'minimal view' with no further details. However deeper in the product catalog, many more details of the TVs will be needed. This implies developing multiple "views" of each cognitive category, selecting different sets of attributes and actions for each. This is an optimization problem using the same guidelines mentioned above: we strive to always present definitional attributes, and we rank other attributes by utility. Additionally, when there are multiple views we need to always provide an affordance for going from one view to another, e.g., to expand or collapse details. This preserves the "identity" of the cognitive category, and makes the user feel they are dealing with something they understand well.

As mentioned earlier, one important attribute of any cognitive category is its "Part-of" relationship to other cognitive categories. Recall the Lady Gaga example in the

Introduction, which baffled users with a strange mental model. As we remarked there, a very simple test would have detected the errors and prevented them. That example violates the structure connecting cognitive categories.

To see this, perform the following exercise: scrunch up your eyes until the picture looks like a bunch of blobs, and call out what objects you see. The first object that stands out is a Story, about Lady Gaga and the video message. The next one is the 'List of Stories', about Lady Gaga and John McCain, etc.

Then ask the question: what is the relationship between these two objects? A Story is *Part-Of* a List of Stories. Does the picture communicate that? No, the List of Stories is inside the box depicting the main Story; the *Part-Of* relationship has been inverted. The list should have been outside, and the highlighted story inside the list.

As one structurally correct solution, the designer could have implemented these relationships by using an accordion widget for the list, showing one article expanded. Clicking on another article would expand it in-place, preserving the *Part-Of* relationships.

Clearly it is easy to detect these errors just by inspection once we develop sensitivity to the types of objects and their relationships to each other. And it's almost as easy to come up with alternative correct designs by mimicking the correct structure.

This point about just following the underlying structure of the cognitive categories has huge implications. E.g., it implies that efforts to "design the "navigation" of a site are misplaced. Rather, navigation should be viewed a byproduct of cognitive category based design. The actual design task is to ensure that all the attributes of an object can be accessed (at least indirectly) no matter where it appears. The user will then automatically follow the relationships between objects (the links) and `good navigation' will be a side effect.

The Visual Design Phase. By the time we reach this step we have established the items that the user needs to interact with, and we have captured the particular meaning and actions that the users associate with those items. Now we need to visually render a design that reinforces that meaning and makes the content instantly recognizable.

A visual designer we worked with once described his job aptly that of a costume designer ensuring that an actor on stage is recognizable no matter which scene they are appearing in and no matter how significant a role they are playing. The actor may change their garb and change their behavior in many ways; yet not so much that the audience struggles to recognize them in different scenes. Likewise a designer needs to ensure that the design instantly brings out the identity of the objects present in the experience, whether they appear in a full or more compressed view. While this advice seems intuitive and obvious, the numerous commonly found violations to this principle attest to the difficulty in adherence.

Consider for example the following illustration of three types of computers from a product catalog (Fig 3a). Note that the visuals for desktops and workstations are rather similar; in fact in testing we observed that customers had difficulty determining which of the two they wanted. This is not an accident; the violation of the image marker rule predicts that users would have a hard time distinguishing the two products. In contrast, note that servers have a very different visual; the monitor is missing, and the image is cropped. The cropping neatly communicates scale; servers are apparently bigger than desktops and workstations. Users were much more easily able to tell whether they wanted a server or a desktop.

Fig. 3. A. Selecting a product category B. Adding a comment in Google Docs

An example from Google Docs (Fig 3b.) illustrates another point about deriving the visual design from the intrinsic nature of the object. We have often heard users complaining that they entered comments into a Google Doc, but now can't find the comments. Upon investigation it turns out that they never hit the "Comment" button on this widget after entering the text. They assumed that since the yellow box was present on the page and didn't vanish when they clicked out of it, that it was saved. In truth the widget does not save comments unless the "Comment" button is clicked.

The root of the problem is that "comment" is both a noun and a verb. The former is a cognitive category, and the widget correctly provides an object-like feel of a yellow sticky note on the page. The verb "to comment" is an action. It's an action performed by the user—it's not an action performed on the Comment object. The actual action performed on the Comment object is "to save". The button should therefore have been labeled "Save". By labeling it "Comment", the design tacitly blinded the user to the fact that it was an action since the *object* comment was already in their head. It didn't help that the button visuals were also changed; if both buttons were gray, there might have been a consistent cue that all actions are gray buttons, instead of one of them appearing to be a different type of entity. The inconsistent switching of visual cues about what constitutes an action versus what constitutes an object, compounded by the dual role of the word "comment" laid a trap for the unwary user.

This highlights the importance of understanding the nature of content in terms of the logic of cognitive categories, and systematizing visual design rules according to the nature of those objects and actions.

We discovered a powerful little trick to help us determine the best visual to use for any given object: just run an image search. I.e., go to a major search engine such as Yahoo, enter the name of the object and click on the Images filter. We are presented with numerous pictures matching the search query. Frequently most the images are quite similar to each other, or fall into a few groups of similar images. Whatever is common about these images defines the object.

For example, a designer recently posed to us the challenge of how to visually represent a “favorite location”. They were designing a map-based application and planned to use the standard upside-down teardrop as the symbol for a location. However they did not know how to mark some of those teardrops as `favorites’.

We ran an image search for “favorite” and promptly discovered that two symbols are frequently used: the star and the red heart sign. That instantly gave us the clues we needed: one solution would be to insert a star inside the teardrop. The other solution would be to add a notch into the top the teardrop making it begin to resemble an elongated heart—and color it red.

Either way, we are exploiting the fact that cognitive categories have distinctive visuals in our brains, and search engines often pick up on those relationships by virtue of crunching tons of user-generated data from the real world.

4 Conclusions

Current research in neuroscience continues to yield fascinating insights into the nature of the human mind. In this paper we have illustrated how some of those insights, from cognitive category theory, can be transformed into powerful tools for designers.

When utilized within the framework of a systematic end-to-end design methodology as we described, it becomes very easy to build user experiences that are certain to be highly intuitive and useful. We now use these methods routinely in our design work and are frequently delighted by how much value we get in just minutes from comparing the structure of our design to the structure of the user’s mental model.

References

1. Norman, D.A.: The design of everyday things, 1st Basic paperback. Basic Books, New York (2002)
2. van der Veer, G.C., del Carmen Puerta Melguizo, M.: The Human-computer Interaction Handbook. In: Jacko, J.A., Sears, A. (eds.), pp. 52–80. L. Erlbaum Associates Inc., Hillsdale (2003)
3. Mental Models and User Experience Design, `http://www.nngroup.com/articles/mental-models/` (accessed: February 20, 2014)
4. Young, I.: Mental models: Aligning design strategy with human behavior. Rosenfeld Media, Brooklyn (2008)
5. Craik, K.J.W.: The nature of explanation. Cambridge University Press, Cambridge (1967)
6. Johnson-Laird, P.N.: Mental Models. Harvard University Press (1983)
7. Gentner, D., Stevens, A.L.: Mental Models. Psychology Press (1983)
8. Lakoff, G., Johnson, M.: Philosophy in the flesh: The embodied mind and its challenge to Western thought. Basic Books, New York (1999)
9. Eco, U., Santambrogio, M., Violi, P.: Meaning and Mental Representations. Indiana University Press (1988)