Syntagm Limited

Why Metaphor is a Double-Edged Sword (Adapted from *interactions* May/Jun 2000)

Don Norman is not one to steer clear of controversy. In *The Invisible Computer* he writes of his disapproval of metaphor in the design of user interfaces – an opinion that he has repeated in debate on the CHI-WEB email list.

What's all the fuss about? The original issue in CHI-WEB was whether to use a shopping cart metaphor when designing e-commerce sites. Some contributors pointed with glee to examples of shopping sleds and wheelbarrows, while others insisted that a cart is a cart is a cart. Don said that we shouldn't be using any of these things and should just have "lists of items purchased". Naturally, this left quite an air of confusion over the whole subject. To try to iron things out, metaphorically speaking, we need to have a better understanding of some of the issues.

Metaphor attempts to express a new domain (the target) in terms of one that is already understood (the source). This is quite common in language where we discuss *negotiation* as if it were war ("he stood his ground"), use the term *higher* to mean *more* and view life as a journey (as in "where do you want to be in two years?"). Some psychologists even argue that thought and language are fundamentally metaphoric. Unfortunately this model of metaphor is very complex, requires a good understanding of both the source and target domains and is very specific within culture and language. An approach that suits user interface design better is called the *structure mapping theory of analogy* and is the work of Dedre Gentner and her colleagues. This theory still relates target and source domains, but by considering the relationships between the conceptual objects (concepts) within each.

The Controversy

What are the objections to metaphor in user interface design? The most commonly quoted are that metaphor is not helpful except to inexperienced users and that it is overly restrictive. The "desktop" metaphor, which is the basis of most graphical environments found today, is often cited as an example of the failure of metaphorical design. However, I think it deserves a closer look. Figure 1 shows a conceptual model of the source domain – an office of the mid 1970's. The rectangles represent concepts while the connecting lines are relationships. I have drawn this in the style of a UML class model so that concepts and relationships are read from the labeled end of each line. So, for example, *in-tray receives document* and *filing cabinet stores folder*. This model (with some minor changes, such as *printer* instead of *copier*) was the basis of the desktop metaphor used on the Xerox Star, the inspiration for the Apple Macintosh, Microsoft Windows and other graphical environments. The trays, folders, documents and other concepts were represented with clear icons and descriptive text, allowing inexperienced computer users to rapidly learn the use of a complex and innovative computer system.

First appeared in *interactions*, Volume 7 , Issue 3 (2000), pages 11-15, Association for Computing, New York, NY.

Notice, by the way, that Figure 1 includes a relationship that really isn't of the same type as the others: *desktop covered by blotter*. While this relationship was certainly true of many desktops at the time, most of the other relationships are concerned with actions. The desktop relationship with blotter is superficial and not part of the system of relationships that interest us. This is called a non-systematic relationship.

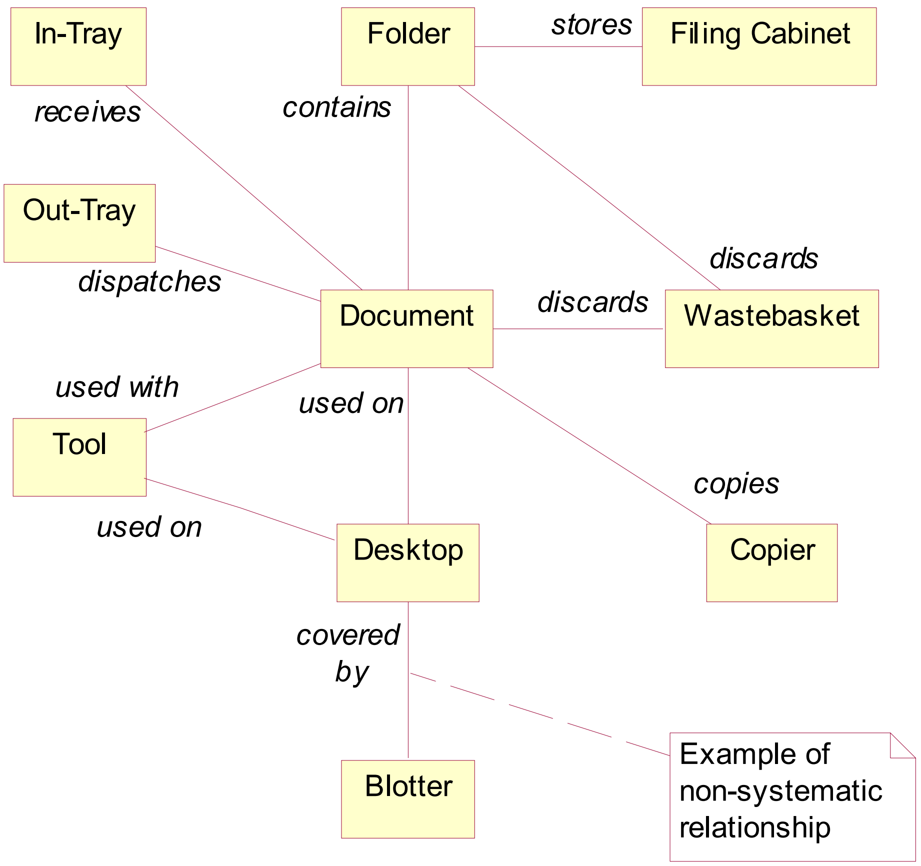

Figure 1, Conceptual model of a mid-1970's office

But why is the desktop metaphor unhelpful? I think the desktop metaphor on our computer screens has drifted too far from the original office domain. Microsoft Windows, for example, hides the in-tray and out-tray in an email application, variously called Mail, Inbox, Outlook or Outlook Express depending on the age and configuration of the system. The *desktop* is confusingly covered by *wallpaper* and printing documents by dragging them to a printer does not work reliably. The Macintosh does not fare much better and also has the odd feature of allowing users to drag the floppy disk to the wastebasket ("Trash") in order to eject the floppy. From the office domain it's obvious that the effect of that action (if any) should be the same as discarding a document or folder. So in most cases, the apparent problem with the desktop metaphor is that it does not correspond to the original office domain.

Now to the suggestion that metaphors confine their designers and users. This supposes that by establishing a well-understood system of relationships we are limited to *just* those relationships. Although it is likely that inexperienced users will understand the metaphor very literally, this same understanding will give them the confidence to explore and experiment. As long as we do not introduce anomalous relationships such as *wastebasket ejects floppy*, there will still be considerable freedom to innovate. For example, within the desktop metaphor, we could make use of a stapler to group and compress documents (as compared to the now common zipper concept which is not part of the office domain). We can also use *bridging* concepts to move from one metaphor to another. A screwdriver (Figure 2) could be used as a bridge from the desktop metaphor to the physical hardware. More advanced users might be able to drag the screwdriver to the filing cabinet to perform maintenance operations such as modifying partitions or defragmenting storage. All of these metaphorical design features could be in addition to more direct (but less predictable) mechanisms such as popup menus and shortcut keys.

Figure 2, Bridging concept from desktop to hardware

Some Problems

Having defended the general concept of metaphor, let's be honest about its limitations. The first problem is that designers sometimes try to be too literal in their use of metaphor. For example, in a real office, mailing a document to a customer means that you no longer have a copy of it, unless you have explicitly made a copy beforehand. Enforcing this behavior in our virtual office would be incredibly frustrating to users. Mindless consistency is not an attractive design strategy.

A larger difficulty is that there really aren't that many useful metaphors for completely new problem domains. The most successful metaphors are those which "virtualize" an existing problem domain. This means adopting not only the concepts and relationships of the domain, but also the activities, as shown for a "virtual store" in Figure 3. Other candidates for virtualization are libraries, clubs, medical services, automotive sales and service – in short, most things that have well-defined domains.

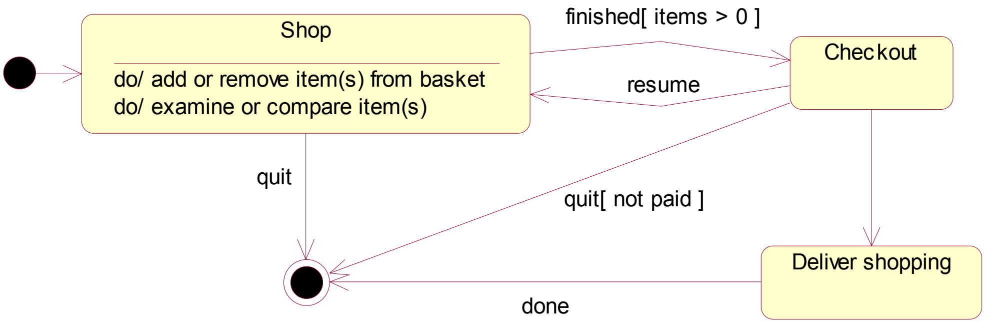

Figure 3, Activity (state) diagram for a virtual store

The Name is the Thing

The debate over terms *shopping sled* versus *shopping cart* may seem relatively trivial at first sight. It is easy to forget that to inexperienced web shoppers, a shopping cart may be a familiar comfort. Arriving at an e-commerce site that does not use the shopping cart metaphor may be a learning experience, rather than a shopping one. It could also be a wasted learning experience if most other web sites use different concepts and activities. The novelty will be beneficial only if it offers a real improvement over familiar solutions *and* is adopted by the Internet community as a whole.

An Iceberg Model of Metaphor

Many concepts are domain specific. For example, a cash register is almost always associated with shopping and the activity of paying. Other concepts, such as money are much more general. The image of money in an e-commerce site may be associated with paying, but it may mean pricing, discounts or currency. The difference is that using a domain-specific concept such as a *cash register* invokes the system of hidden relationships to which it belongs. The cash register may be the only visible evidence of the metaphor, but the rest of the shopping domain is lurking beneath the surface.

This iceberg model of metaphor can present problems as well as providing solutions. A number of web sites have taken to using the term *check in*. Some use it as an activity required of visitors to the web site (i.e., registering or joining), while others ask existing members to check in. Which is right? The term *check in* is used almost exclusively in the travel industry to mean registration for a pre-arranged journey or stay and so is associated with a reservation as shown in Figure 4. The question of whether the customer is a visitor or a member does not really arise, except that a member might be tenuously considered to have a reservation. It would be best not to use the term at all outside its travel context.

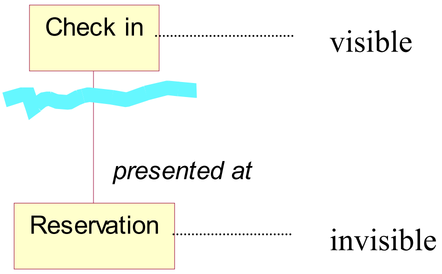

Figure 4, *Check in* has a *reservation* hidden below the surface

Metaphoric Guidelines

My own view is that metaphor is an important tool for user interface design, but must be used with care. The guidelines below cover some of the most important issues:

- Metaphors operate on systems of relationships, not on individual concepts. Make sure that the system of relationships is reflected in your user interface and do not use concepts out of context.

- Metaphors do not have to be complete, but interfaces need to provide adequate clues to users. Omitting less important concepts or changing them slightly may not have a large impact on usability, but only testing will tell.

- Metaphors should not rely on mere appearance. A concept should not just have the appearance of a shopping basket, it should behave like one too. It needs to have the same or similar relationships in the target domain as it did in the source.

- Avoid non-systematic relationships (such as *desktop covered by blotter* in Figure 1). Most of the important relationships will be how concepts interact with each other, particularly from a user's perspective.

- Don't let abstract relationships interfere with the metaphor. For instance, it may be true that a desk and a filing cabinet are both instances of office furniture. However, it's their purpose that is of interest.

- Choose metaphors that provide concrete images. The shopping domain is useful because it contains a number of domain-specific concepts that can easily be represented visually: shopping cart (or basket, trolley, etc.), cash register, price tags, discount signs, etc. A catalog metaphor, by comparison, is relatively devoid of distinct concrete images (an order form is not as immediately recognizable as a cart or basket).

- Try to get as close to the original domain as possible (unless an alternative has obvious advantages). Mail-order shopping and vending machines are already one step removed from the source shopping domain. Also, some aspects of the vending machine domain are not entirely beneficial to e-commerce: Vending machines usually sell low value items, most require payment in advance, and not all are reliable.

- Beware culture-specific metaphors and concepts. *Cart* (US) versus *trolley* (UK) is a minor issue since the image is the same. However, *kiss and ride* (dropping off a loved one at a purpose-built public transport facility) does not enjoy wide-spread recognition.

Further Reading

Gentner, Dedre and Markman, A. B. (1997), 'Structure mapping in analogy and similarity', in *American Psychologist,* Vol. 52, pp. 45-56.

Ortony, Andrew, ed. (1993), *Metaphor and Thought*, Cambridge University Press, Cambridge.

The Author

William Hudson is a freelance software designer based near Oxford in the UK. His experience ranges from firmware to desktop applications, but he started by writing interactive software in the early 1970's. For the past ten years his focus has been user interface design, object-oriented design and HCI.

William Hudson
Syntagm Ltd
Design for Usable Systems
Tel +44 (1235) 522859
Fax +44 (1235) 554449
Email mailto:whudson@syntagm.co.uk
Web http://www.syntagm.co.uk
vCard http://www.syntagm.co.uk/whudson.vcf