

The Death of Metaphor

Andrew Kowalczyk

October 31, 2013

Metaphor, in the broadest sense of the term for the discipline of a computer science, is a group of user interface images and activities that take advantage of particular proficiencies that users already have of other domains. Typical examples include using a tool or some kind of gear to represent Settings or Preferences on some operating system or “flipping” a page in a book reading app. Although the majority of people designing user interfaces of any sort employ metaphors heavily today, I argue that the beginning of the end of metaphor is well on it’s way.

1 Introduction

Although this phenomenon is not extremely widespread just yet, hints of the oncoming change can be seen everywhere. Metaphor is starting to wane.

In terms of how long humans have been roaming this earth, electronic technology is quite a new thing for our two-legged species. Within the last 50 years, hard drive capacities have exponentially grown while their physical space has exponentially shrunk. Seeing that technology changes nearly on a daily basis, it seems that it would be logical for computer scientists to employ the concept of metaphor. For nontechnical users, it would make sense to employ powerful metaphors which simulate real-life. One would assume that adding more life like design into the way someone interacts with a given piece of technology would increase their overall usability. In terms of creating a usable interface, it is funny to think that the way in which we interact with that technology hasn’t changed all too much. In the examples that I mentioned in the abstract, users know that tools can modify things in real life, the connection is clear for any user utilizing that said operating system.

Imagine a computer scientist thinking about how they could design a user interface that works well with people who are not technical? Use what they already know.

Metaphor limits the designer because it gives less freedom to employ there way they do things. Metaphor limits the user because it disallows for true learning.

[1]

2 Background, Prior Work, and Literature Read

In this section, I go over the literature I read to support my claims and how they relate to my argument.

2.1 Background

2.2 Literature Read

2.2.1 Apple's push away from metaphor

With no surprise, the release of iOS 7 garnered much support and much backlash. Many of the large criticisms include its transition to overly simple menus, its change from icons to words, and its unlikelike apps.

Apple has realized that as technology is becoming more and more ingrained into our society, the aid of metaphor is largely shrinking.

2.2.2 Minimalism in Computer Science

Minimalism is a very sought out concept in computer science. Code golf (writing programs with as little characters as possible),

2.2.3 Dijkstra's argument

Edsger Dijkstra, one of the most important computer scientists, claimed that metaphor was extremely damaging to the profession even before the age of personal computers and the rise of GUI's. He initially introduces his argument by explaining the way in which we approach tomorrow's struggles. He says the most common way of facing tomorrow is by "means of metaphors and analogies" so that "we try to link the new to the old, the novel to the familiar". He makes the claim that coping with novelty in this fashion is extremely detrimental for a number of reasons.

First, he does admit that in the cases of "sufficiently slow and gradual change", metaphor works reasonably well. A powerful example is language. The process by which word's meaning evolve is a slow and gradual process. Yet, in the case of "sharp discontinuity", the employment of metaphor breaks down. metaphor becomes more "misleading than illuminating". It makes sense, doesn't it? In a field that changes nearly daily, the relevance of our

[2]

2.2.4 Summary of literature read

3 Methods

4 Discussion

4.1 Views

5 Conclusions

References

- [1] Saskia Bakker, Alissa N. Antle, and Elise van den Hoven. Embodied metaphors in tangible interaction design. *Personal and Ubiquitous Computing*, 16:433–449, 2012.
- [2] Edsger W. Dijkstra. On the cruelty of really teaching computer science, 1988.