Metaphor is a rhetorical technique through which one concept is expressed in terms of another. For example, to say "that party was a nightmare" is not to literally describe a nightmare about a party, but rather to leverage what a listener already knows about nightmares against what he doesn't yet know about the party. As discussed in detail by Aristotle in *Poetics*[f-poetics], this equation of concepts is a common device in literature.

Analysis of this topic goes beyond the use of metaphor simply as a figure of speech and into the areas of thought and action. Of particular interest here is conceptual metaphor, an idea examined in detail by George Lakoff and Mark Johnson in *Metaphors We Live By*. Discussed in that book is Lakoff and Johnson's technique of mapping between target and source domains, where the target is the *party* concept to which the explanatory *nightmare* concept is mapped. This kind of reasoning seems to be pervasive in human thought, and plays a prominent role making sense of the world.

Firmly rooted in cognitive linguistics, the principles of conceptual metaphor are useful in studying the interaction between computers and their users. Lakoff and Johnson take a category-based approach to the analysis of metaphor, and this will serve as inspiration for the following discussion of spatial metaphor, personification and temporal metaphor. Each category will be presented with analysis of real-world examples, and this will be followed by discussion of the arguments for and against the use of such principles in the design of computer systems.

In computer interface design, perhaps the most commonly invoked of these categories is spatial metaphor. The intangible functions of software are often notionally presented as physical spaces, objects or actions. The personal computer offers many such examples, with the desktop metaphor being the de facto standard basis on which modern operating systems are designed. In this schema, the computer monitor represents the user's physical desktop space, within which documents and folders can be opened, moved and layered. Software applications, when opened, may also be presented on the desktop in a way that is consistent with the metaphor. In Microsoft Windows, for example, Word gives access to a WYSIWYG[f-wysiwyg] view of a document as if it were laid out on the desktop, and Calculator presents a calculator application much as it would look on the physical desktop.

As part of the desktop metaphor, digital resources are presented in the context of files and documents, and a hierarchical structure for browsing and managing these files is established as folders. This is roughly analogous to accessing documents in a real filing cabinet. The notion of an office environment is further extended with functionality such as trash cans for unwanted files. Icons provide the visual interface for all of these elements, and often in themselves feature visual metaphor. For example, an icon for accessing data on a disk may depict the disk's physical form, or an icon to launch a text editing application may depict a pencil and a piece of paper.

Access to functionality above the desktop tends to be through windows, which simply act as containers for on-screen content. Windows can be layered on top of and tiled adjacent to each other, duplicated, moved and resized. They are also accompanied by menus, which allow users to open resources or launch applications. Beyond the operating system, the functions available at the application level also tend to be metaphorical, often being presented as real-world tools such as *pencil*, *paintbrush* and *lasso*. Other tools reference actions, such as Adobe Photoshop's *crop*, *burn* and *dodge*.

At this point it is worth noting that everything mentioned so far emerges from some very basic principles, themselves examples of metaphor. The basic unit of communication within a computer is a binary unit, or bit. In practical terms this is a pulse of electricity

that, when sent across the computer's circuits, denotes an *on* state. This is analogous to flicking a switch on a mechanical device, and is the foundation on which more advanced logic can be built. However, this is a rather complex way of considering the whole setup, and so bits are often abstracted to the idea of 0s and 1s. An interesting series of object - and, specifically, food - metaphors then emerges as the principle is extended. A collection of two bits is known as a crumb, of four bits is known as a nybble and of eight bits is known as a byte. At a higher level still, programming languages begin to emerge, the functions and variables of which are in themselves merely abstractions of the more complex logic at work beneath them.

Unlike a PC, the World Wide Web is distributed and intangible, and as such has had a great many metaphors applied to describe its nature as a whole. The idea of a web is in itself metaphorical, having been chosen by the Web's inventor, Tim Berners-Lee, as a way of conveying the idea of easily navigable, interconnected resources[f-web]. It can be described as a library, with books representing websites and pages representing webpages. A highway metaphor is also prominent, with terms such as *information superhighway* establishing the Web as a conduit for high-speed, high-bandwidth, bidirectional information transfer. The term cyberspace is also commonly used to present activity within the Web in a spatial context[f-cyberspace]. Even Lego metaphors have emerged to describe the Web as an array of interoperable, interlocking technologies and resources[f-lego].

The actions associated with Web browsing are also of interest here. Users might be said to be surfing, grabbing files, delving into content or hitting a website. Browser software such as Mozilla Firefox allows pages to be recalled with *back*, *forward* and *home* actions, implying movement through a physical space. Orientational metaphors such as *uploading* and *downloading* imply movement between the user's machine and another, as well as indicating the hierarchical importance of each.

As self-contained interfaces, websites predictably have many established metaphors of their own. The ubiquitous shopping cart has a literal basis in simply storing a list of items the user wants to purchase, but setting this in a physical context can be seen to benefit the user experience[f-shopping_cart]. Even on a first visit to the site, a user will recognise that the cart allows them to review items, remove them and change quantities. This representation of the intended transaction is notionally taken to the checkout, another concept sourced from real-world shopping. Other common interface elements such as the guestbook, as well as more specific ideas such as the *Wall* feature of social networking site Facebook[f-facebook_wall], use metaphor to similar effect.

Deserving of a special mention in this discussion of spatial metaphor is the lexicon not of computer interfaces themselves, but of their enthusiasts. Words used by the hacker and cracker communities, for example, shed interesting light on the concepts at play in the use of these systems. The term *hack* maps an aggressive physical action to the idea of manipulating hardware or software. The verb *crack*, in this context, maps a similarly aggressive action to a user's attempt to overcome digital authentication or copy protection measures. Terms like *phishing* and *pharming*, which denote fraudulent online activities[f-phishing_pharming], give conceptual clues about the activity by referencing the physical actions of fishing and farming. Other relevant terms include *patching*, *trolling* and *flaming*.

A development of the spatial metaphor is characterisation, where interface elements are imbued with sentience. This is, in a broad sense, personification, but may be expressed through non-human forms such as animals or robots.

In interacting with a personified interface, a user may be afforded instant understanding of the system's behaviour through their existing knowledge of the character's behaviour. Web search engine Ask Jeeves, for example, was designed around the idea of using a valet character to mediate natural language and keyword-based Web searches. In presenting the technology in this way, the user's familiarity with a valet's role could be exploited. Instantly conveyed is the idea of service, whereby commands can be issued and results expected. It could also be argued that choosing a 'high-class' character of this type makes for an aspirational experience, and has the advantage of implying diligence and attention to detail.

Sentient beings have also appeared prominently in Microsoft's Office suite of applications. Here, a range of animated Office Assistants has been used to deliver help tips and take simple user input. For example, Microsoft Word might monitor a document as it is being written, and if it looks like a letter, pop up a paperclip character to offer a selection of help topics on effective letter-writing. The user can dismiss the character or follow the options provided to access a more comprehensive help document.

Interestingly, the design of each Office Assistant highlighted an inconsistent approach to the implementation of this metaphor. The appearance of some characters was indicative of their metaphorical source. Word 97's Will character, for example, has a clear likeness to William Shakespeare, and thus an ability to provide help in producing quality writing can be inferred. It is hard to extract any useful information at all, however, about what Hoverbot, a hovering robot, or The Dot, a red ball with a cheery expression, can offer. Having said this, it seems possible that the notion of *animated character as help interface* is sufficiently prominent in its own right to provide a functioning metaphor, irrespective of how the character is drawn.

References to a character may be more oblique than those mentioned so far. Something found in many applications is a wizard[f-wizard], a step-by-step dialogue that solicits information from the user to solve a particular task. This device accesses the user's notions of what a mythical wizard can achieve. Results can be conjured up, seemingly appearing out of nowhere and with minimal effort. The reference has in the past been explicit, with graphical representations of wizards or magic wands, but is increasingly implicit, with no such visual cues. An implicit approach can be said to be exhibiting the characteristics of dead metaphor, whereby it has been so heavily used as to have lost its figurative value. The source notion of a mythological wizard is no longer required because the target notion has become so firmly established in its own right.

Although not actually represented as sentient, other digital phenomena are considered in a very similar light. To say that a computer virus has "taken on a life of its own" is very telling, as it implies a certain level of self-awareness. The virus is programmed with needs and desires, often attempting to find new hosts, access certain information locally and communicate with other machines.

It should also be noted at this stage that the term computer can in itself be considered metaphorical, and is another example of personification. Alan Turing's *On Computable Numbers, With an Application to the Entscheidungsproblem* highlights how, in the early days of computing, the term *computer* referred to human operatives[f-turing]. Real people would be responsible for solving mathematical problems, and would do so by sitting down with a pencil and paper, and manually processing information. It was only later, as mechanical information processing developed, that *computer* became useful in this new context. In being transferred in this way, the new and unfamiliar territory of mechanical computing was expressed as being an extension of an established practice, and thus

metaphorical significance was evident. Such comparisons between human and mechanical computing tie in closely with personification and the ideas of mind-as-computer and computer-as-mind[f-mind_computer].

The final and most specific metaphorical category to be discussed is temporal metaphor, and particularly that which is technologically regressive in nature. For a metaphor to be of use, its source domain must be familiar, otherwise there would be no value in equating it to the unfamiliar target domain. The examples discussed so far have generally found this familiarity outside of technology, sourcing simple concepts like paper, desks, libraries and roads from the physical world. What is interesting in the design of computing interfaces is that this approach is often subverted, and source domains are chosen that are relatively close to their target. The name of web-based streaming video service YouTube, for example, establishes a conceptual link to cathode ray tube televisions that were ubiquitous until just a few years ago. This approach instantly conveys to the user an idea of what the site offers and how it might be used.

Sitting down to watch a traditional television broadcast is, of course, very different to actively browsing a website for video clips. Television is a *push* medium offering high bandwidth and high fidelity, but with very limited flexibility. Web video offers lower bandwidth and lower fidelity, but is an asynchronous, interactive and user-focused experience[f-nielson]. These differences are irrelevant, however, in the context in which the metaphor is applied, so the equation of the two concepts works as intended.

Technologically regressive references may be even more subtle than this. Music comparison service Last.fm is available online at http://last.fm, the *.fm* part of which is the country code top-level domain for the Federated States of Micronesia. This might seem strange as the company is based in London, England, but is an example of a domain hack, where the top-level domain is chosen for a reason other than the country it represents. In this case, a reference is being made to FM radio broadcasting in an attempt to convey some of the characteristics of this medium. A user can access a variety of music streams, building personal radio stations based on recommendations. The process is entirely Internet-based, but radio is a useful concept in summing up the functionality of the site.

What is interesting in these regressive metaphors is that the conceptual differences between source and target are not nearly as great as with some of the physically-derived metaphors discussed previously. Sources such as TV or radio are relatively complex notions in themselves, and would have required their own metaphors to be understood properly at the time of their own inception. For example, the 60s saw the television cited variously as an X-ray machine for cultural analysis and a mirror held up to the behaviour of the public[f-tv].

A key argument against the use of metaphorical techniques in digital design is that physically-derived metaphor brings with it a great deal of baggage, limiting the potential of the systems in which it features. Based on bits, rather than atoms, computer systems are freed from the constraints of the physical world. In practical terms, bits take up no space and can be accessed, replicated and destroyed instantly at the user's will. A system based on subatomic particles is inherently intangible, and is therefore interacted with very differently than the physical world. With this in mind, the application of physical metaphor might be identified as a weak substitute for finding a good conceptual representation of the computer system in its own right. Ted Nelson discusses this concern with reference to the paper paradigm in Dream Machines:

> It's very amusing. People sit at computer screens and think they're working with paper. This

> paper paradigm, this paperdigm, permeates today's computer world. But the notion of paper - a two-dimensional, sequential relation of facts and ideas - is one of the things most holding back software design and human progress.
>
> Today's computer systems simulate paper in stupefying detail, enshrining it as if tomorrow would never come, as if today's frozen methods and concepts were permanent.
>
> Paper is a crutch, an old-fashioned idea that is holding us back. When screen methods get good enough - and they are far from it now - there will be no more hard copy. And it is toward that day we must design - toward finer, more flexible, all-seeing methods that we could never have in these last paper days.
>
> Eventually we must drop the ancient ways and frozen methods, we must throw away this security blanket of paper, we must let go of the side of the pool, and find those new virtualities of tomorrow's hypermedia that will free our minds and unchain our work.

There are many examples of ill-fated metaphor use in computer design. One commonly cited example is MacOS's implementation of the *trash* icon. As well as being a location to which unwanted files can be dragged, the icon has the less obvious function of ejecting disks in the same way. Consistency would dictate that the disk's files be deleted by doing this, but the disk is merely ejected. This seems a counter-intuitive approach, and one in which the underlying desktop metaphor is proved problematically inflexible. A broken and obtrusive metaphor results.

In other cases, the integrity of the metaphor may be rigorously maintained to the detriment of the interface. The wizard paradigm mentioned previously often proves to be an example of this because it imposes a rigid, linear structure on information input. Questions such as "what do you want to do?", "what name do you want to use?" and "what options do you want to use?" may be useful if the user hasn't experienced the process before, but an experienced user will most likely desire more of a freeform approach. Being forced to enter data in such a granular fashion can be extremely tedious.

Badly implemented metaphor caused much greater damage to Microsoft Bob, a software package released in 1995 intended to provide a non-technical interface for novice computer users. The product was meant as an alternative to the desktop metaphor, instead providing functionality in a cartoon-like simulation of a 3D environment. Characters such as Rover, a yellow dog, and Digger, an Irish earthworm, mediated this interaction through speech bubbles and wizards. Bob is a user interface crippled by an overly rigid implementation of a spatial metaphor.

Another criticism of metaphor is that it can degrade over time. The paper paradigm, for example, is one that has clearly become less relevant since it first saw use in computers like Xerox PARC's Star desktop. Owing to poor screen technologies, the desire to output to paper greatly affected interfaces of the day. In 2007, significant progress has been made towards paperless office environments, with consumer products such as Amazon's Kindle e-book reader offering considerable advantages over paper-based reading.

Developing systems which move away from metaphor and its constraints has interested HCI experts for many years, and although the desktop metaphor and paper paradigm remain dominant, there are systems that show higher levels of conceptual purity. Such systems are arguably more faithful to Ted Nelson's notion of virtuality, a term describing the experience that software alone, without reference to the physical world, offers.

As the use of metaphor seems to be tied up in an interface's shortcomings just as much as

it is those of the user, it is no suprise that many attempts at more literal design have featured some radical departures in hardware and interface. Microsoft's Surface project, a multi-touch[f-surface] system allowing direct interaction with digital content through physical gestures and natural motion, is a recent example. The system more closely represents how a user might interact with real-world objects because the on-screen content can be grabbed, pulled, stretched and rotated just as if it were made of atoms. By establishing direct contact in this way, interactions become much more like those in the real world, and metaphor becomes less prominent as the intermediary between human and computer.

The alternative to rejecting metaphor as a design paradigm is to recognise both its strengths and its weaknesses and work with them. Indeed, the spatial, character-based and temporal metaphor discussed previously are not conceptually pure. The designers of the interfaces have been somewhat adventurous in selectively extending, mixing or even discarding metaphor in order to streamline the user experience, and these principles can be taken much further. The Microsoft Windows taskbar is one such example, in that it exists quite apart from the desktop on which it is located. An non-metaphorical element like this has no physical precedent and could only exist in software, and thus is an interesting realisation of virtuality.

Alternatively, metaphor can be sustained in the system overall but discarded completely in certain areas. One prominent example is the desktop search functionality in use on operating systems such as Windows Vista. Nested within the traditional desktop and menu schema, the feature allows the input of text, which can be used to locate files or launch applications. Accessing things in random and flexible ways like this is not a principle that translates to the physical world, and is therefore significantly beyond the scope of the desktop metaphor.

Despite the many criticisms of metaphor, its use in computer interfaces persists. Its presence is unsurprising because it is holistic in nature, and is ingrained in computing from the lowest to the highest level of complexity. Its prominence is also unsurprising because computing hardware still lacks the finesse to present purer virtuality in an intuitive and user-friendly way. Consequently, the number of users willing or able to interact with this sub-optimal hardware without the aid of metaphor is very small. As long as these conditions exist, it will be necessary to explicitly load interfaces with metaphor. Only in this way can the concepts of the system and the abilities of the user be reconciled. This is a transitional, best-fit solution, but one which has given a robust solution for creating inviting and accessible systems. Thanks to the familiar concepts conveyed, even the uninitiated user can evaluate the system easily and identify a viable learning curve. The ubiquity of such systems in the current marketplace can no doubt be attributed to this.

The mainstream interfaces of 2007 show that metaphor can be used effectively. A computer system can be presented coherently on the basis of multiple metaphors, and these metaphors are remarkably malleable. Different metaphors within the same system can be mixed and extended way beyond their original scope. As with the paper paradigm, a single metaphor may feature prominently, but a mere essence of the source concept is often all that is needed to convey the necessary principles. The compromise lies in giving enough metaphorical cues to appease novices while leaving the underlying target concepts sufficiently unconstrained to make the system flexible, powerful and unobtrusive.

The metaphors in use will of course evolve over time. As improved hardware makes interaction with paper documents a thing of the past, the paper paradigm will cease to act as a meaningful source domain. [...]

Footnotes

[f-poetics] Aristotle. 1967. *Poetics*, Michigan: University of Michigan Press. pp. 57

[f-wysiwyg] WYSIWYG is an acronym for *what you see is what you get*, and describes an editing interface that shows a prediction of what the finished content will look like. Software such as Microsoft Word, for example, shows an on-screen representation of what the printed page will look like.

[f-web] Berners-Lee, Tim. 1999. *Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor*, London: Orion Publishing. pp. 8 - 27

[f-cyberspace] Coyne, Richard. 1995. *Designing Information Technology in the Postmodern Age: From Method to Metaphor*. Cambridge, Massachusetts: MIT Press. Coyne offers a detailed discussion of the phenomenology of cyberspace between pages 147 and 177.

[f-lego] As reported at http://www.news.com/8301-10784_3-6076301-7.html.

[f-shopping_cart] Stephanidis, Constantine et al. 2003. *Human-Computer Interaction: Theory and Practice*. New Jersey: Lawrence Erlbaum Associates. pp. 1188 - 1189

[f-facebook_wall] The Wall is a default application applied by Facebook. Displayed on each user's profile page, it serves a location for leaving publicly-viewable messages.

[f-phishing_pharming] Phishing refers to the act of fraudulently acquiring personal information by masquerading as a trustworthy entity. For example, a user may be asked to enter their banking details at an illegitimate page designed to resemble the bank's login page. Pharming is one process by which this redirection of the user is achieved.

[f-wizard] According to a report by Microsoft at http://www.microsoft.com/presspass/press/2001/oct01/10-15TenYearsPublisherPR.mspx, the wizard appears to have first been implemented in Microsoft Publisher in 1991.

[f-turing] Turing, A. 1936. *On Computable Numbers, With an Application to the Entscheidungsproblem*. pp135 - pp140

[f-mind_computer] Boden, Margaret. 2006. *Mind as Machine: A History of Cognitive Science*, Oxford: Oxford University Press. pp. 176 - 179

[f-nielson] Web usability expert Jakob Nielson discusses these differences at http://www.useit.com/alertbox/9705b.html.

[f-tv] Bernhard, Nancy. 1999. *U.S. Television News and Cold War Propaganda, 1947-1960*. Cambridge: Cambridge University Press. pp 51 - 55

[f-surface] Technical details are available at http://www.microsoft.com/surface/.

Bibliography

Coyne, Richard. 1995. *Designing Information Technology in the Postmodern Age: From Method to Metaphor*. Cambridge, Massachusetts: MIT Press
Johnson, Mark. 1987. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*, Chicago, Illinois: University of Chicago Press
Kaptelinin, Victor and Mary P. Czerwinski. 2007. *Beyond the Desktop Metaphor: Designing Integrated Digital Work Environments*, Cambridge, Massachusetts: MIT Press
Lakoff, George and Mark Johnson. 1980. *Metaphors We Live By*, Chicago, Illinois: University of Chicago Press
Nelson, Theodore. 1987. *Computer Lib/Dream Machines*, London: Tempus