The Pleasure Principle

Robert R. Hoffman and Patrick J. Hayes, *Institute for Human and Machine Cognition*

The list of "concepts that psychology really can't do without" includes such notions as neuronal connectionism, degrees of consciousness, mental representation of information, and dissociation. Of the pantheon of contributors to the history of psychology, Aristotle outranks all others in terms of the number of critical concepts he introduced, including the notion of the association of ideas, the law of frequency and the affiliated concept of memory strength, the notion of stage theories of development, the idea of distinguishing types of mental processes or faculties, the idea of scales of nature and comparisons between humans and animals, and last but not least, the Pleasure Principle.

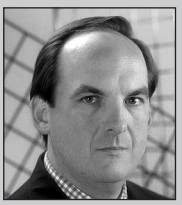
History of the Pleasure Principle

In his *Physics*, Aristotle wrote, "All moral excellence is concerned with bodily pleasures and pains."[1–3] What he was getting at is that animals as well as humans experience pleasure and pain but that a human who lets such factors alone direct his or her behavior would be intemperate, impetuous, brutish, and self-indulgent to excess. He further said, "[In] the case of bodily enjoyments … the man who pursues excessive pleasures and avoids excessive pains like hunger and thirst, heat and cold, and all the discomforts of touch and taste, not from choice but in opposition to it and to his reasoning, is described as incontinent [driven to excess by an uncontrollable appetite] without any added determinant [or cause of behavior]."[2] Humans are distinguished from mere brutes by having within them "a rational principle," but being the animals that we are, we still share the universal Pleasure Principle.

Based on his studies of trial-and-error learning in animals, Edwin Thorndike proposed a variant[4] of the principle: "Any act in a given situation producing satisfaction becomes associated with that situation, so that when the situation recurs, that act is more likely to recur," recapitulating Aristotle's notion of association and his law of frequency. This general idea for a causal explanation of behavior in terms of affect was so critical and useful that even the Behaviorists such as John Watson thought they could embrace it in a theory devoid of all the cooties of mentalism, referring to behaviors that get "stamped in" because they are "reinforced" and behaviors that get "stamped out" because they lead to punishment.[5] In Sigmund Freud's work,[6] which also echoed many Aristotelian notions, the principle was transformed to the more familiar "Humans behave so as to seek pleasure and avoid pain." This was the force of the id, in contrast to the "reality principle" that governed the ego. In the early 1900s, the often-misunderstood "efficiency experts" (such as Frank Gilbreth[7]) were likewise cognizant of the principle. They wanted to increase worker productivity—not just for its own sake but also to eliminate wasteful work practices and increase worker health and psychological satisfaction.

The Pleasure Principle remains a theme today. Kim Vicente subtitled his opus on cognitive work analysis *Toward Safe, Productive and Healthy Computer-Based Work,* implying that psychological satisfaction is an ingredient in "health."[8] Job satisfaction is a critical factor in determining worker morale, productivity, and health. When working either as individuals or as team members, people show higher levels of emotional investment in their projects, greater levels of commitment, greater staying power in the face of impediments, and higher levels of accomplishment if the group perceives itself to be functioning effectively.[9] So, as Aristotle pointed out, pleasure makes even rational work more effective.





Editors: Robert R. Hoffman, Patrick J. Hayes, and Kenneth M. Ford
Institute for Human and Machine Cognition, University of West Florida
rhoffman@ai.uwf.edu

1094-7167/04/$20.00 © 2004 IEEE
Published by the IEEE Computer Society

How does this history lesson map to human-centered computing?

Unfortunately, computers don't always provide an unmixed increase in pleasure. Recent evidence suggests, contrary to what we might hope or suppose, that the computerization of the modern workplace has actually led to productivity declines.[8,10]

The negative impacts are likely due, at least in part, to the user unfriendliness of computers—for example, frequent changes in software, incompatibility of hardware and software systems, poor software and interface design, weak documentation and help support, and so on.[10] Many computer systems require effort to create work-arounds and local kludges.[11] All such features make both individual workers and teams feel less effective, and perceived self-efficacy is a critical factor in job satisfaction, motivation, and morale.[9]

Examples

Examples abound, but here's an example from Studs Terkel's classic book *Working*, in which he interviews a telephone operator:

> Half the phones have a new system where the quarter is three beeps, a dime is two beeps, and nickel is one beep. If the guy's in a hurry and he keeps throwing in money, all the beeps get mixed up together (laughs) and you don't know how much money is in the phone. So it's kinda hard. When you have a call, you fill out this IBM card. Those go with a special machine. You use a special pencil so it'll go through this computer and pick up the numbers. It's real soft lead, it just goes all over the desk and you're all dirty by the time you get off.[12]

Here's a second example to which we can all relate, this from newspaper columnist Dave Barry:

> The next day I booked another flight to Chicago ... I was flying with a ticket that said my name was "Barry White." Really. That is who the airline computer insisted I was. I pointed out to the ticket agent that Barry White is a famous soul crooner and does not resemble me in any way except that we are both bipeds. I asked if my ticket could reflect my real name; after tapping on his computer for a good ten minutes, the agent informed me—I swear—that this was not possible, and advised me to just get on the plane.[13]

A third clear example is one that arose during our research on HCC for weather forecasting.[14] NEXRAD (the Next-Generation Weather Radar system) is a marvel of technology, with capabilities yet to be fully explored. The NEXRAD Principal User Processor (PUP) workstation has the slick feel of *2001: A Space Odyssey*, but its appearance is misleading. Operation relies on a graphics pad that is anything but self-explanatory. Its colored sectors have functional significance, but that significance is totally hidden in the dozens of cryptic acronyms, abbreviations, and alphanumeric encodings that label the individual buttons. The user interface is a command line interface, requiring the operator to be familiar with dozens of commands, coding schemes, and so on. Operations manuals are always kept well within reach.

NEXRAD begs for both a graphical user interface (GUI) and a knowledge-based support system. It certainly need not suffer from an outdated assumption that the way to pack information into an interface is to abbreviate and encode. In the months after NEXRAD was installed (with initial enthusiasm) at the US National Weather Service's weather forecasting offices, the PUP went largely unused at some WFOs that one of us (Hoffman) visited. In some of those offices, the PUP still collects dust, and the forecasters work with NEXRAD products at their separate forecasting workstations. Only after the WFOs' science officers took NEXRAD training did the regional WFOs start really using the system (defining local thresholds, setting up special user functions, and so on). Even after that, WFO forecasters would refer to their one NEXRAD-trained colleague as their local "guru," meaning that only one forecaster was actually good at working with the interface, interpreting the NEXRAD products, and adapting the radar's operating characteristics to local weather regimes.

> Others are often quite happy to work with a "guru" while they themselves almost proudly remain a "not-guru"—hence terms such as propeller-head, geek, hacker, turbine guy, and so on.

During one of the interview sessions in the Weather Case Study project, the following conversation took place:

> Interviewer: NEXRAD is so capable but there's no knowledge … . From a user point of view it's like having a Cadillac with all the bells and whistles but on the dashboard there are no labels, or the labels are like hexidecimal code, cryptic commands ….
>
> Forecaster: A lot of times you do not need to know how it works [which is primarily what is taught in the schoolhouse], you need to know how the radar reacts to things, different atmospheres … actually how to use it. [But] it really doesn't invite you to have fun with it, no. Not at all. It's nice they have all the user functions … the more you use it the more it invites you to use it. You can set up [your own] user functions.

This makes the point exactly. Once you start to use NEXRAD, it gets better and better because you can customize it. But it's too hard to get started. The forecaster must engage in experiential learning for a considerable time. This is unfortunate because NEXRAD has such fantastic capability. A workshop on NEXRAD human factors was held at the 1997 Meeting of the American Meteorological Society, and discussions at that workshop showed the promise of NEXRAD as an effort began to develop object-oriented GUIs. Until then, most NEXRAD users rely on one or two basic displays of the radar data, which they view at their workstations rather than at the NEXRAD PUP.

Being the guru

There is an additional subtlety here. Being the "guru" can be pleasurable. Others are often quite happy to work with a "guru" while they themselves almost proudly remain a "not-guru"—hence terms such as *propeller-head*, *geek*, *hacker*, *turbine guy*, and so on. So, these complex "usable-only-with-a-great-deal-of-training" systems actually can give pleasure but not in the way their designers had in mind. They encourage a kind of two-level workplace that is apparently stable socially but that can be inefficient in exactly the way the systems are not supposed to be—that is, they fail to provide user-friendly functionalities. No doubt, this is in part because of the designer-centered approach in which interfaces are designed by engineers who often have exactly the kind of personality that would have taken the training course and taken pleasure from playing the guru role. In extreme cases (we've all met these folks), they can be almost unaware of the existence of not-guru users, or maybe

aware but contemptuous. This too illustrates the dangers of social instability that can make systems effectively useless in a socially healthier work situation.

Good tools self-explain to the greatest extent possible and don't assume that information must be crammed into data fields (at the expense of communicating meaning). Good tools make tasks neither so unnecessarily difficult as to instill hopelessness or frustration, nor so easy as to promote users' boredom or anger.[15] Good tools (and workspaces) motivate the worker. They don't involve distractions or disruptions (for example, having to look something up in a paper manual) that destroy the subjective experience of being "in the domain." They don't force the worker to create work-arounds and spend hours merely learning how to use the tool rather than getting the actual job done.

Are you having fun?

The role of emotion and aesthetics in HCC was raised a few times in the 1997 National Science Foundation workshop report on human-centered computing.[16] Pelle Ehn pointed out that quality in the workplace, like quality of life in general, necessarily involves aesthetic considerations.[17] Ross Jeffries defined HCC in terms of three characteristics:[18]

- The system does something that people want and need to do.
- The system is well integrated into real practice.
- The user is able to focus on the task rather than on the user interface.

But then Jeffries went a step further and added a fourth characteristic:

> I'll add to this a characteristic that may be somewhat controversial, but I have come to see as an important aspect of human-centered systems. The system should be fun to use. It's easy to see fun as an extra, an add-on (or even something frivolous, to avoid). By fun I don't mean "joke of the day" features or MTV-like presentations, although in the right context either of these could be a good idea. Rather, I mean that using the system leaves the user in a better state of mind than before. I don't know if fun is something we have to explicitly design into our systems, or if it is an emergent property of being sufficiently human-centered, but I have come to see it as an essential property of successful systems. (p. 277)

Here, Jeffries converges with Donald Norman on a cardinal principle of HCC, which we have formulated as a modern version of the Pleasure Principle:

> Good tools provide a feeling of direct engagement, flow, and challenge.

As we struggled to reach an integrative view of HCC,[19–21] we realized that something was missing from many discussions of such things as "user friendliness." When the Principle dawned on us, it felt dangerous at first. We took that as a clue that it was indeed important to information technology and intelligent systems. Since then, we've heard others allude to similar notions and now feel more comfortable discussing it.

This involves creating intelligent systems that let practitioners work problems rather than having to work their technologies in order to work problems.

HCC isn't about turning every complex sociotechnical workplace into a fun place to be, although it would be good if that happens where and when it's appropriate and possible. A retired senior naval officer commented to one of us (Robert Hoffman), "There ain't *no* place on board a ship where you *want* to go." This comment captures the fine line that must be walked in formulating the Pleasure Principle. "Fun" in the sense that we and Jeffries intend doesn't mean ho-ho-ho. It's more a feeling of engagement, an *integration* of Aristotle's animus with the rational process. You need them both—or perhaps better, the rational can't do nearly so well by itself if it has to do without the pleasure part.

The point of the Pleasure Principle is that human-centered systems must leverage domain practitioners' intrinsic motivation, especially the motivation that is definitive of expertise. This involves creating intelligent systems that let practitioners work problems rather than having to "work" their technologies in order to work problems. Indeed, the more important the job, the more important it is that the work environment let practitioners feel engaged in working toward a goal, experience no frustrations, and then feel satisfied that they have successfully, effectively, and effortlessly achieved their goal.

How the Pleasure Principle relates to other HCC principles

The Pleasure Principle relates to other HCC principles, including the following:[22]

- *The Sacagawea Principle*. Human-centered computational tools need to support active organization of information, active search for information, active exploration of information, reflection on the meaning of information, and evaluation and choice among action sequence alternatives.
- *The Lewis and Clark Principle*. The human user must be guided in a way that's organized in terms of his or her major goals. Information needed for each particular goal should be shown in a meaningful form and should allow the user to directly comprehend the major decisions associated with each goal.

Both are suggestive of a state in which practitioners are directly perceiving meanings and ongoing events, experiencing the problem they are working or the process they are controlling. The challenge is to live in and work on the problem, not to have to always fiddle with machines to achieve understanding.

Anyone practicing a skill has experienced this: musicians playing, dancers dancing, drivers driving, carpenters carpentering, even engineers engineering. But when it comes to the complex sociotechnical workplace, it's the rare piece of software that makes fiddling unnecessary for practitioners or that makes fiddling better for people who like to fiddle (unless of course fiddling happens to be programming or some bastardized version of it—for instance, when "work-around" really means "reprogramming").

The tool should let you see your problem or work better, not force itself on your attention. The tool "becomes part of you," as good tool users often say. This is the HCC point in a nutshell. The connection with the Pleasure Principle is that people in fact take pleasure from using a skill well and that a good tool should amplify and utilize this;

but a bad tool forces you to become skilled at using the tool. The tool becomes the subject matter. So nobody is better at doing the original job, but now we have a new kind of job (using the bloody tool!) that requires a new kind of skill. That can become kind of fun for some people, but it's not the actual job. Meanwhile, the original job is no more fun or any easier than it used to be.

The Pleasure Principle should be pushed for all it's worth, perhaps even to the point of being considered a criterion in requirements analysis and the procurement process. Aristotle, we assume, would see that as virtuous. ■

Acknowledgments

The authors' contribution to this article was made possible by participation in the Advanced Decision Architectures Collaborative Technology Alliance, sponsored by the US Army Research Laboratory under cooperative agreement DAAD-19-01-2-0009.

References

1. Aristotle, *Physics*, Book 7, Section 3, W.D. Ross, ed., Clarendon Press, 1908.
2. Aristotle, *Nicomachean Ethics*, W.D. Ross, trans. and ed., Clarendon Press, 1908.
3. G. Irbe, *Aristotle's Nicomachean Ethics*, 2000, www.interlog.com/~girbe/home.html.
4. E.L. Thorndike, "Animal Intelligence: As Experimental Study of the Associative Processes in Animals," *Psychological Rev. Monograph Supplements*, no. 2, 1898.
5. J.B. Watson, "Psychology as the Behaviorist Views It," *Psychological Rev.,* vol. 20, 1913, pp. 158–177.
6. S. Freud, *Civilization and Its Discontents*, standard ed., J. Strachey, ed., W.W. Norton, 1961.
7. F.B. Gilbreth, *Motion Study*, Van Nostrand, 1911.
8. K. Vicente, *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*, Lawrence Erlbaum, 1999.
9. A. Bandura, "Exercise of Human Agency through Collective Efficacy," *Current Directions in Psychological Science*, vol. 9, 9 July 2000, pp. 75–78.
10. T.K. Laudauer, *The Trouble with Computers*, MIT Press, 1997.
11. P. Koopman and R.R. Hoffman, "Work-Arounds, Make-Work, and Kludges," *IEEE Intelligent Systems*, vol. 18, no. 6, Nov./Dec. 2003, pp. 70–75.
12. S. Terkel, *Working*, New Press, 1972, p. 37.
13. D. Barry, "There Is No Plane, but Your Flight Is Still on Schedule," *The Pensacola News J.*, 9 July, 2000, p. 4G.
15. D.A. Norman, *Things that Make Us Smart*, Perseus Books, 1993, p. 35.
16. J. Flanagan et al., eds., *Human-Centered Systems: Information, Interactivity and Intelligence*, US Nat'l Science Foundation, 1997.
17. P. Ehn, "Seven 'Classical' Questions about Human-Centered Design," *Human-Centered Systems: Information, Interactivity and Intelligence*, J. Flanagan et al., eds., US Nat'l Science Foundation, 1997, pp. 267–269.
18. R. Jeffries, "Position Paper," *Human-Centered Systems: Information, Interactivity and Intelligence*, J. Flanagan et al., eds., US Nat'l Science Foundation, 1997, pp. 277–279.
19. R.R. Hoffman, K.M. Ford, and J.W. Coffey, "The Handbook of Human-Centered Computing." a deliverable on the Human-Centered System Prototype contract, Nat'l Technology Alliance, 2000; available from Robert Hoffman, rhoffman@ai.uwf.edu.
20. R.R. Hoffman, P.J. Hayes, and K.M. Ford, "Human-Centered Computing: Thinking In and Out of the Box," *IEEE Intelligent Systems*, vol. 16, no. 5, Sept./Oct. 2001, pp. 76–78.
21. R.R. Hoffman et al., "A Rose by Any Other Name … Would Probably Be Given an Acronym," *IEEE Intelligent Systems*, vol. 17, no. 4, July/Aug. 2002, pp. 72–80.
22. A. Endsley and R. Hoffman, "The Sacagawea Principle," *IEEE Intelligent Systems,* vol. 17, no. 6, Nov./Dec. 2002, pp. 80–85.

Robert R. Hoffman is a research scientist at the Institute for Human and Machine Cognition. Contact him at the IHMC, 40 Alcaniz St., Pensacola, FL 32501; rhoffman@ihmc.us.

Patrick J. Hayes is a senior research scientist and John C. Pace Jr. Eminent Scholar at the Institute for Human and Machine Cognition. Contact him at the IHMC, 40 Alcaniz St., Pensacola, FL 32501.